Advant® OCS

AdvaInform® Basic Functions

User’s Guide

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaCommand, AdvaInform, AdvaTalk and AdvaBuild are registered trademarks of ABB Asea Brown Boveri Ltd, Switzerland.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrial Systems AB ABB Industrial Systems AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrial Systems AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrial Systems AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrial Systems AB’s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

CE MARKING

This product meets the requirements specified in EMC Directive 89/336/EEC and in Low Voltage Directive 72/23/EEC.

Copyright © ABB Industrial Systems AB 1997

3BSE001264/D

Template: 3BSE001286/E

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 General Information 1-1
1.2 Equipment Requirements 1-2
1.3 How to Use This Book 1-2
  1.3.1 Manual Organization 1-2
  1.3.2 Differences Between Master and MOD 300 1-3
1.4 Conventions 1-4
1.5 Related Documentation 1-5
1.6 Release History 1-8
1.7 Terminology 1-10
1.8 Product Overview 1-14
  1.8.1 Operating System 1-15
  1.8.2 The AdvaInform Relational Database 1-15
  1.8.3 Web Support 1-16
  1.8.4 Communication 1-19
    1.8.4.1 Advant OCS Control Network/DCN Communications . 1-19
    1.8.4.2 Stand-alone Configuration 1-20
    1.8.4.3 Configuration 1-20
    1.8.4.4 Plant Network Communications 1-20
    1.8.4.5 Communication Gateways (Master) 1-22
  1.8.5 User Interface and Menu Systems 1-23
    1.8.5.1 IMS Menu 1-23
    1.8.5.2 System Status 1-26
    1.8.5.3 Object Presentation Access 1-27
    1.8.5.4 Terminal Window 1-28
    1.8.5.5 Mail 1-28
    1.8.5.6 System Messages 1-28
    1.8.5.7 Editor 1-28
  1.8.6 Alarm & Event Handling 1-28
  1.8.7 System Administration 1-30
    1.8.7.1 User Definition 1-30
    1.8.7.2 Daylight Savings and Time Synchronization 1-30
    1.8.7.3 Station Backup 1-31
  1.8.8 AdvaInform Application Development Tools 1-31
  1.8.9 Support for Optional Software 1-32

CONTENTS (continued)

Chapter 2 - Installation

2.1 Site Planning and Environment 2-1
2.2 Setup, Start up and Shut down 2-1

Chapter 3 - Configuration/Application Building

3.1 Design Considerations 3-1
3.2 Capacity and Performance 3-1
3.3 Application Start-up and User Authorization 3-1
  3.3.1 User Authorization 3-1
  3.3.2 HP-UX Users 3-4
  3.3.3 Log In Procedure 3-5
  3.3.4 Log Off Procedure 3-7
  3.3.5 Log In to the AdvaInform Relational Database 3-8
  3.3.6 Oracle users 3-8
  3.3.7 IMS Menu Icon 3-9
3.4 Tutorial 3-9
3.5 Application Procedures 3-10
  3.5.1 Application Building within AdvaInform Basic Unit 3-10
  3.5.2 Access to AdvaInform Functionality from an Operator Station 3-10
    3.5.2.1 Available Displays 3-10
    3.5.2.2 Master Operator Station 3-12
    3.5.2.3 MOD 300 Operator Station 3-13
3.6 Menus 3-13
  3.6.1 AdvaInform Menu 3-13
    3.6.1.1 AdvaInform History: TTD Configuration (Master) 3-13
  3.6.2 Settings Menu 3-15
    3.6.2.1 Settings - Communication 3-16
    3.6.2.2 Settings - User Definition 3-16
    3.6.2.3 Settings - Station Backup 3-16
    3.6.2.4 Settings - Process Supervision 3-17
    3.6.2.5 Settings - Maintenance 3-17
    3.6.2.6 Settings - System Variables 3-17
    3.6.2.7 Settings - System Messages 3-17
    3.6.2.8 Settings - RTA (Master) 3-17
    3.6.2.9 Settings - Date and Time 3-17
    3.6.2.10 Settings - Node Broker (Master) 3-17

CONTENTS (continued)

Chapter 4 - Runtime Operation

4.1 Product Operation 4-1
4.2 Operating Overview 4-2
4.3 Runtime Tutorial 4-4
4.4 Operating Instructions 4-4
4.4.1 How to Manage the CD-ROM reader 4-4
4.4.2 How to Execute Queries in the Database Access Window 4-5
4.4.2.1 General 4-5
4.4.2.2 How to List All Tables, Views and Object Types for a User 4-6
4.4.2.3 How to List all Attributes and Columns for a Table or View 4-6
4.4.2.4 How to Query for a Numerical Log Object 4-6
4.4.2.5 How to Query for a Message Log Object (MOD 300 example) 4-8
4.4.2.6 How to Present Objects and Logs 4-10
4.4.3 How to Use System Status 4-14
4.4.3.1 System Overview Display 4-18
4.4.3.2 Node Status Display IMS 4-20
4.4.3.3 Node Status Display Advant Controller (Master) 4-21
4.4.3.4 Node Status Display - Master Nodes 4-22
4.4.3.5 Node Status Display Operator Station (Master) 4-23
4.4.3.6 Network Display (Master) 4-24
4.4.4 How to Access the Relational Database 4-25
4.4.5 How to Edit 4-25
4.4.6 How to Send Mail 4-26
4.4.7 How to Analyze System Messages 4-26
4.4.8 How to Resolve Object References (Master) 4-29
4.4.8.1 Open Gateway 4-32
4.4.8.2 Retranslate Object References 4-34
4.4.8.3 Find Object 4-36
4.4.8.4 List Object References 4-38
4.4.8.5 Set Restart Option 4-44
4.4.9 How to Show Current User 4-45
4.4.10 How to Change Language 4-46
4.4.11 How to Change Password 4-47

CONTENTS (continued)

4.5 Runtime Menus 4-47
  4.5.1 File Menu 4-47
    4.5.1.1 File - Exit 4-47
  4.5.2 Station Menu 4-48
    4.5.2.1 Station - System Status 4-48
    4.5.2.2 Station - Object Presentations 4-49
    4.5.2.3 Station - Terminal Window 4-49
    4.5.2.4 Station - Database Access 4-49
    4.5.2.5 Station - Editor 4-50
    4.5.2.6 Station - Mail 4-50
    4.5.2.7 Station - System Messages 4-50
    4.5.2.8 Station - Object Reference Maintenance (Master only) 4-50
  4.5.3 AdvaBuild Menu 4-51
    4.5.3.1 AdvaBuild - Object Type Builder 4-51
    4.5.3.2 AdvaBuild - Central Backup (Master only) 4-51
    4.5.3.3 Programmer 2000 - Pro*C 4-51
  4.5.4 AdvaInform Menu 4-52
    4.5.4.1 AdvaInform - Object Handling 4-52
    4.5.4.2 AdvaInform - SQL*Connect 4-52
    4.5.4.3 AdvaInform - User API 4-53
    4.5.4.4 AdvaInform - History 4-53
    4.5.4.5 AdvaInform - Calculations (Master only) 4-53
    4.5.4.6 AdvaInform - Reports 4-53
    4.5.4.7 AdvaInform - Event Log (Master only) 4-53
    4.5.4.8 AdvaInform - SPC (MOD 300) 4-54
    4.5.4.9 AdvaInform - PDL (MOD 300) 4-54
  4.5.5 AdvaTalk Menu 4-54
  4.5.6 Session Menu 4-54
  4.5.7 Help Menu 4-55

Chapter 5 - Maintenance

5.1 Preventive Maintenance 5-1
5.2 Hardware Indicators 5-1
5.3 Presenting System Messages 5-1
  5.3.1 Oracle *oerr* on-line tool 5-2
  5.3.2 Message Box 5-2
  5.3.3 IMS Menu 5-3

CONTENTS (continued)

5.4 System Messages 5-3
 5.4.1 Alarm and Event Distributor - Process bcaedistr.... 5-3
  5.4.1.1 Fatal error messages.... 5-3
  5.4.1.2 Error messages 5-4
  5.4.1.3 Warning messages.... 5-5
  5.4.1.4 Information messages 5-6
 5.4.2 System Status display (Master).... 5-7
  5.4.2.1 Fatal error messages.... 5-7
  5.4.2.2 Warning messages.... 5-7
  5.4.2.3 Information messages 5-8
5.5 Fault Finding and Repair 5-9

Chapter 6 - FAQ (Frequently Asked Questions)

Appendix A - Relational Databases, SQL, and SQL*Plus

A.1 Relational Databases....A-1
A.2 SQL (Structured Query Language)....A-3

Appendix B - SQL Statement Reference

B.1 General....B-1
B.2 CREATE DATABASE LINK B-1
 B.2.1 Purpose....B-1
 B.2.2 Prerequisites....B-1
 B.2.3 Syntax B-1
 B.2.4 Keyword and Parameters....B-1
 B.2.5 Usage Notes B-2
 B.2.6 Example B-2
 B.2.7 Related Topics....B-3
B.3 CREATE VIEW....B-3
 B.3.1 Purpose....B-3
 B.3.2 Prerequisites....B-3
 B.3.3 Syntax B-3
 B.3.4 Keywords and Parameters B-3
 B.3.5 Usage Notes B-4
 B.3.6 Naming Guidelines B-4
 B.3.7 AUDITING VIEWS B-5
 B.3.8 Examples....B-5
 B.3.9 Related Topics....B-6

CONTENTS (continued)

B.4 DESCRIBE (Dynamic Embedded SQL: Method 4) ... B-6
  B.4.1 Purpose ... B-6
  B.4.2 Prerequisites ... B-6
  B.4.3 Syntax ... B-6
  B.4.4 Keywords and Parameters ... B-6
  B.4.5 Usage Notes ... B-6
  B.4.6 Example ... B-7
  B.4.7 Related Topic ... B-7
B.5 SELECT ... B-7
  B.5.1 Purpose ... B-7
  B.5.2 Prerequisites ... B-7
  B.5.3 Syntax ... B-7
  B.5.4 Keywords and Parameters ... B-7
  B.5.5 Usage Notes ... B-8
  B.5.6 CONNECT BY ... B-9
  B.5.7 START WITH ... B-9
  B.5.8 GROUP BY, HAVING ... B-9
  B.5.9 Set Operators ... B-9
  B.5.10 ORDER BY ... B-10
  B.5.11 FOR UPDATE OF ... B-10
  B.5.12 Sequences ... B-10
  B.5.13 Subquery ... B-10
  B.5.14 Examples ... B-10
  B.5.15 Related Topics ... B-10
B.6 SELECT (Embedded SQL) ... B-11
  B.6.1 Purpose ... B-11
  B.6.2 Prerequisites ... B-11
  B.6.3 Syntax ... B-11
  B.6.4 Keywords and Parameters ... B-11
  B.6.5 Usage Notes ... B-12
  B.6.6 Examples ... B-12
  B.6.7 Related Topics ... B-13

CONTENTS (continued)

B.7 SELECT STATEMENT CLAUSES B-13
  B.7.1 FROM clause B-13
    B.7.1.1 Purpose B-13
    B.7.1.2 Prerequisites B-13
    B.7.1.3 Syntax B-13
    B.7.1.4 Keywords and parameters B-14
    B.7.1.5 Usage Notes B-14
    B.7.1.6 Examples B-14
    B.7.1.7 Related Topics B-15
  B.7.2 WHERE clause B-15
    B.7.2.1 Purpose B-15
    B.7.2.2 Prerequisites B-15
    B.7.2.3 Syntax B-15
    B.7.2.4 Keywords and Parameters B-15
    B.7.2.5 Usage Notes B-16
    B.7.2.6 Examples B-16
    B.7.2.7 Using the TO_DATE Function in a Condition B-17
  B.7.3 WHERE clause (Embedded SQL) B-19
    B.7.3.1 Purpose B-19
    B.7.3.2 Prerequisites B-19
    B.7.3.3 Syntax B-19
    B.7.3.4 Keywords and condition Parameters B-20
    B.7.3.5 Usage Notes B-20
    B.7.3.6 CURRENT OF Clause B-20
    B.7.3.7 Example B-20
    B.7.3.8 Related Topics B-20
B.8 UPDATE B-21
  B.8.1 Purpose B-21
  B.8.2 Prerequisites B-21
  B.8.3 Syntax B-21
  B.8.4 Keywords and Parameters B-21
  B.8.5 Usage Notes B-22
  B.8.6 Examples B-23
  B.8.7 Related Topics B-24

CONTENTS (continued)

B.9 INSERT B-24
  B.9.1 Purpose B-24
  B.9.2 Prerequisites B-24
  B.9.3 Syntax B-24
  B.9.4 Keywords and Parameters B-24
  B.9.5 Usage Notes B-25
  B.9.6 Examples B-25
  B.9.7 Related Topics B-25
B.10 DELETE B-26
  B.10.1 Syntax B-26
  B.10.2 Example B-26
  B.10.3 Related Topics B-26

Appendix C - Object Presentations

C.1 AI Object Presentation C-1
C.2 AO Object Presentation C-5
C.3 DI Object Presentation C-8
C.4 DO Object Presentation C-11
C.5 DAT Object Presentation C-13
C.6 TEXT Object Presentation C-15
C.7 PIDCON Object Presentation (Master only) C-18
C.8 RATIOSTN Object Presentation (Master only) C-25
C.9 MANSTN Object Presentation (Master) C-30
C.10 NUMERIC LOG Presentation C-34
C.11 Local Event Log Presentation (Master) C-40
C.12 MESSAGE LOG Presentation (most applicable to MOD 300) C-42

INDEX

Chapter 1 Introduction

1.1 General Information

This book describes how to configure and use AdvaInform® Basic Functions. AdvaInform Basic Functions is the software platform for AdvaInform, AdvaBuild®, AdvaTalk® and other options that you can install in an Advant® Station with AdvaInform Basic Functions,

The hardware platform for AdvaInform Basic Functions is the Advant Station 500 Series, Figure 1-3. The combination of the AdvaInform Basic Software and the Advant Station form the basic Advant Station Information Management Station (IMS). Many software options are available for the IMS. See the *Advant Station 500 Series IMS User´s Guide* for more information. The AdvaInform software provides information server functionality in the Advant OCS and Plant Network.

Figure 1-1. Example of AdvaInform functions accessed from an X-terminal

The AdvaInform software is optimized for information gathering and manipulation. It gathers, stores, manipulates and presents process and production information to support informed decision making by production management. It is also a software platform for integrating both your applications and vendor-supplied applications into the Advant OCS.

AdvaInform functions also offer open system features that make it possible to integrate information from all parts of the plant. You can access data consistently, whether the data resides in Advant OCS nodes (internal nodes) or external nodes, or whether it is generated by AdvaInform applications.

1.2 Equipment Requirements

The hardware platform for the AdvaInform Basic Functions is the Advant Station 500 Series. The Advant Station hardware is described in detail in the applicable *Advant Station 500 Series Hardware User's Guide*.

If you require the IMS to communicate with other Advant OCS nodes, the optional Real Time Accelerator (RTA) board must be included. See Section 1.8.4, Communication.

1.3 How to Use This Book

1.3.1 Manual Organization

This book is organized in the following chapters and sections, Figure 1-2.

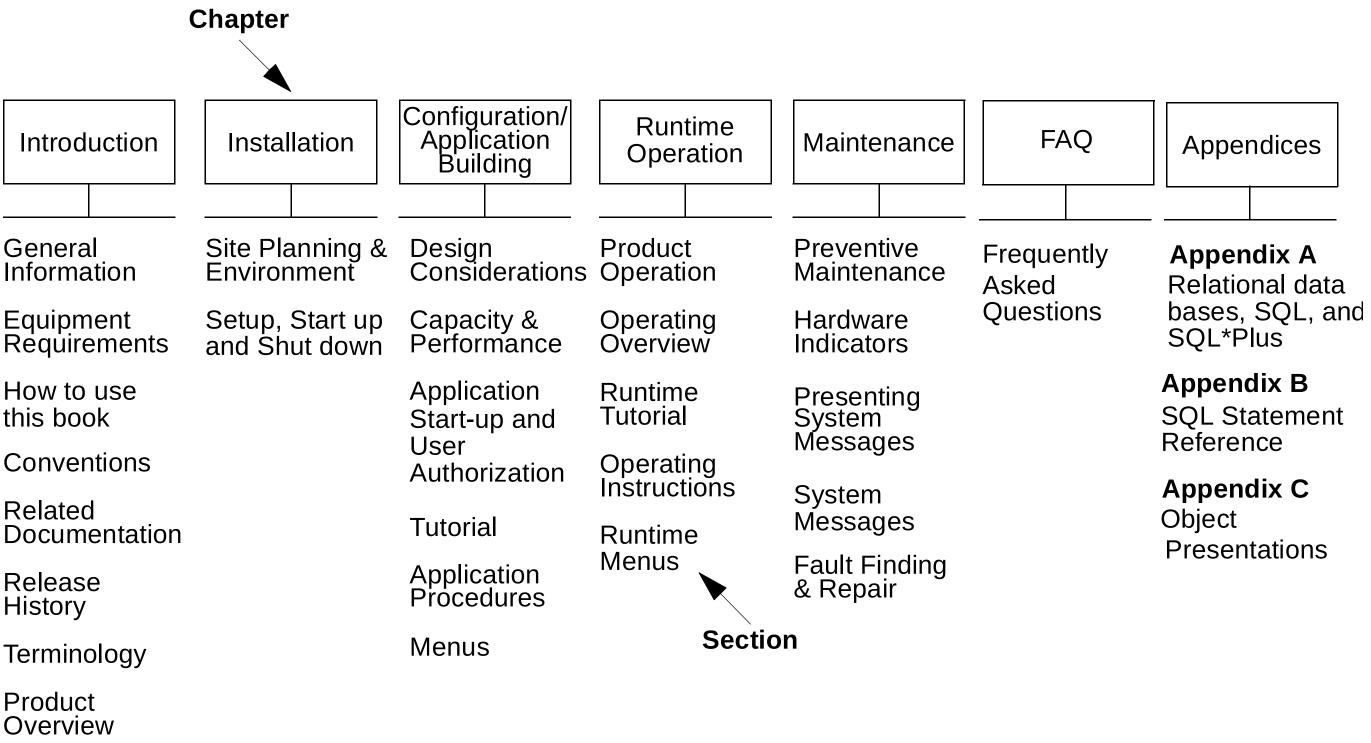


Figure 1-2. Manual Organization

1.3.2 Differences Between Master and MOD 300

The following is a brief summary of the main differences in the application of AdvaInform Basic Functions for Master and MOD 300.

Object Types

While some AdvaInform object types are common for both Master and MOD 300, there are also some object types that are unique to Master or MOD 300. A set of pre-defined Oracle views are available to facilitate data retrieval for these object types.

Some views are common for Master and MOD 300, and some are unique.

Refer to the applicable *AdvaInform Object Type Reference Manual* for details. There are two separate manuals. One for MOD 300 object types and one for Master object types.

MEASURE and VALUE Object Attributes

One common object attribute for MOD 300 objects is the MEASURE attribute. The equivalent attribute for Master applications is VALUE. These attributes are used in some of the examples in this book. In some cases, two examples are provided showing both MEASURE and VALUE, and in other cases just one attribute or the other is shown. In all cases, the examples can be made applicable for both Master and MOD 300 by simply substituting the correct attribute.

Event and Alarm Handling

Event and alarm handling differs for MOD 300 and Master.

TCL and CCF

TCL and CCF are software packages for MOD 300 applications. All examples and references for TCL and CCF are only applicable for MOD 300.

System Status

The System Status of an IMS provides two types of Status presentation and supervision:

- Node Status - provides all relevant status information about the Advant Station and AdvaInform software.
- Network Status - is Master specific and provides information about the connected MasterBus 300 network and the status of the other Advant Master nodes.

AdvaInform Options

Different AdvaInform options are available for MOD 300 and Master. For this reason, there may be different menu items available under the **AdvaInform**, **AdvaBuild and AdvaTalk** menus in the IMS menu bar for MOD 300 and Master.

1.4 Conventions

The following conventions are used throughout this manual for the presentation of material:

- The words in proper names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- The names of keyboard keys are boldfaced.
  - Initial capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, the **Return** key or **<Return>**.
  - Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
  - In descriptions, the word “key” is used in conjunction with its name. For example, pressing the **Return** key initiates the process.
  - In tables and procedures, angle brackets replace the word “key.” For example, press **<Return>**.
  - A dash is used to indicate that you must simultaneously press several keys. For example, **<Ctrl-C>** indicates that you must hold down the **Control** key while pressing the **C** key.
  - The phrase “press and release” is used to indicate that you sequentially press several keys.
- The names of push and toggle buttons are boldfaced.
  - Initial capital letters are used for the name of a push or toggle button labeled on the display. For example, the **OK** button.
  - In descriptions, tables and procedures, the word “button” is used in conjunction with its name. For example, pressing the **OK** button initiates the process.
- The names of menus and menu items are boldfaced.
  - Initial capital letters are used for the name of menus listed in the menu bar. For example, the **File** menu.
  - Initial capital letters are used for the name of menu items. For example, to open a new window, choose **New** from the **File** menu.
- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

  ```
  Entered value is not valid. The value must be 0-30.
  ```

  In a tutorial, you may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure: **TIC132**

1.5 Related Documentation

Table 1-1 lists all documentation related to AdvaInform Basic Functions User's Guide

Table 1-1. Related Documentation

Category	Title	Description
Hardware	Advant Station 5XX Hardware User's Guide	These books cover the installation and maintenance of the hardware for an Advant Station models.
Station	Advant Station 500 Series IMS User's Guide	This book describes the Advant Stations IMS in general. Read this book to find out about the functionality, performance and capacity of all AdvaInform options.
Options	AdvaInform History User's Guide	This book describes how to configure and use the History option.
	AdvaInform Reports User's Guide	This book describes how to configure and execute reports.
	Oracle Developer/2000 Reports 2.5 Building Reports Manual	The Oracle Developer/2000 documentation provides information on Oracle Reports not covered in the AdvaInform Reports User's Guide. This documentation is available from ABB.
	Oracle Developer/2000 Reports 2.5 Reference Manual	
	Oracle Developer/2000 Graphics 2.5 Developers Guide	
	Oracle Developer/2000 Graphics 2.5 Reference Manual	
	AdvaInform User API User´s Guide	This book describes the procedures and reference information for AdvaInform User API.
	AdvaInform SQL*Connect User´s Guide	This book describes the Client and Server parts of AdvaInform SQL*Connect and how to use the available tools to configure those parts.
	AdvaInform SQL*Connect Programming User´s Guide	This book describes the optional SQL*Connect programming option.
	AdvaInform Calculations User's Guide	Master only This book describes how to configure and use the AdvaInform Calculations option.
	AdvaInform Event Log User's Guide	Master only This book describes how to configure and use the AdvaInform Event Log option.

Table 1-1. Related Documentation (Continued)

Category	Title	Description
Options	AdvaInform SPC User's Guide	MOD 300 only This book describes how to configure and use the AdvaInform SPC option.
	AdvaInform PDL User´s Guide	MOD 300 only This book describes how to configure and use the AdvaInform PDL option.
	AdvaInform Object Handling User's Guide	This book describes how to configure and use the AdvaInform Object Handling option.
	AdvaBuild Object Type Builder User's Guide	This book describes how to use the Object Type Builder option.Documentation for C++ compiler which can be ordered from the separate price list for documents, may also be helpful.
	AdvaBuild Central Backup	Master only This book describes how to configure and use the AdvaBuild Central Backup option.
	AdvaTalk Link to MATLAB User's Guide	This book describes how to configure and use the AdvaTalk Link to MATLAB option.
	AdvaInform Display	This book describes how to configure and use the AdvaInform Display options.
	AdvaInform Object Types Reference Manual	This book provides table names and attribute names for all object types supported by AdvaInform. This information is useful when you develop applications using such tools as AdvaInform User API.

Table 1-1. Related Documentation (Continued)

Category	**Title**	**Description**
Other	C/ANSI C Compiler	A C-licence is included in AdvaInform Basic Functions. However, the C manual has to be ordered separately. HP manuals for ANSI-C are needed by the development engineer who uses Object Access Services and uses the C-development license. Documentation includes C language reference manual, C programmers guide, C programming tools, HP-UX symbolic debugger and programming on HP-UX.
	HP C++ Programmer’s Guide	Is recommended for the development engineer who develops applications with AdvaBuild Object Type Builder.
	MasterNet Users Guide	Master only Includes configuration rules for Advant OCS with Master software. It explains the concept of Control and Plant Network and the details regarding time synchronization and daylight saving.
HP-UX 10.20	Read me before installation	An introduction to the installation procedure.
	Quick guide to Installation	A quick guide for the installation procedure.
	Installing HP-UX 10.10 and 10.20	A complete user’s guide for installation of HP-UX 10.20.
	Configure HP-UX peripherals	Includes a description of how to configure different HP peripherals

1.6 Release History

Version	Description
1.0	Initial release of AdvaInform Basic Functions.
1.1	Provides the following new features and enhancements: • Improved performance and stability • Persistent name storage supported. This allows for faster start-up as once symbols (objects) are resolved, they do not have to be resolved again after system start-up. • Full support for Master alarm and event distribution. Alarms and events generated by objects in AdvaInform are presented in OS and MasterView. Alarm acknowledge is also supported. • Master products MasterView and Superview now have full access to AdvaInform Basic Objects. • Database tables for MOD 300 process object types DATA_FCM and VA_STRING_FCM. They are used in conjunction with SQL*Connect Server and they provide access to the AdvaInform relational database from the Advant OCS operator station. • Error logs from the AdvaInform are sent to and presented on MOD 300 diagnostic displays.
1.2	Provides the following new features and enhancements: • Improved performance and stability • IMS 1.2 is based upon Oracle7 and Forms4 • HP-UX 9.05. Supports for example hardware for AS 515/64. • Language support - for english, swedish and german • IMS User Definition support allows the user to define new IMS users. • System Status support for the new Master controllers • Improved installation and start-up. Features to support start-up, shut-down, and supervision of Customer/ABB applications. • Improved workstations backup features, Incremental On-Line backup, on demand and on time schedule. Selective restore functionality. • A tool for maintenance of Object References in a Master system is provided. It allows you to monitor the progress of symbol resolution, open up gateways, resolve symbols after moving objects in your network etc. • New disks (1 and 2 GByte disks, Raid Disks) • C development option (C license) is included in Basic Software.

1.3	Provides the following new features and enhancements: • Distribution of MOD 300 TCL Billboard messages to Operator Stations.
2.0	Provides the following new features and enhancements: • New licensing system. • Based upon HP-UX 10.20 • AdvaInform Basic Software software is installed separate from HP-UX 10.20 • All Advant software is distributed on CD-ROM • Subscription intervals other than 1,3 and 9 seconds for cyclic subscriptions are supported (Master) • The System Status user interface is refreshed and improved • Node Status display available for AdvaInform Basic Functions MOD 300 • User Authorization is improved. The User Definition function allows you to create a unique Oracle user for each IMS/HP-UX user. • The Advant OCS communication is optional. This means that the software and the Advant Station can be installed stand-alone. • Web support is provided through the Netscape FastTrack Web Server and the Netscape Navigator Gold web browser.

For details on new and modified options, see *Advant Station 500 IMS User´s Guide*.

1.7 Terminology

Term	Description
3-GL	Stands for Third Generation Language; a term which refers to languages such as C++ and C.
Advant Controller	A set of different Advant Controllers exist. They handle I/O, control logic, regulation, alarm and event generation etc.
Advant OCS objects	See Object below.
AdvaInform User API	User Application Programming Interface.
API (Application Programming Interface)	A set of subroutines that provide a means to establish remote process-to-process communication. AdvaInform User API in AdvaInform Basic Functions allows programs in the HP-UX environment to interact with Advant OCS objects.
Attribute	The public data of an object. The result of the application and the configured data is commonly stored in an attribute. Other objects can request the object's public data through the Core System using subscriptions. AdvaInform User API applications can request attribute values via the Object Access Services.
Control Network	A network connecting different types of Advant stations/controllers used to supervise and control a specific part of the plant. It is possible to exchange data between a Control Network and the Plant Network and also between different Control Networks via the Plant Network (See Plant Network below).
Command	Operation requests to an object. As a result, the object performs the requested action.
Composite Attributes	Used in User Objects and Basic Objects included in AdvaInform Object Handling. A Composite Attribute is a list of attributes for an Object Type that are available for subscription. (Instead of having to refer to each individual attribute.)
Core System	All nodes in the Advant OCS communicate with each other through the Core System. The Core System provides services for the system applications. Using these services, the system applications can communicate with each other whether they reside in the same node or in different nodes of different types. The Type Directory, which is part of the Core Directory, contains information on all object types known to the Advant OCS.
DCN	Distributed Communication Network. Control network for Advant OCS with MOD 300 Software.
DCS	Distributed Control System
ES	Engineering Station

Term	Description
External Node	An external node is a node that is not part of the Advant OCS. Information Management Stations, Operator Stations, Engineering Stations and Controllers are internal nodes (part of the Advant OCS). An external node can access AdvaInform Basic Functions via the X-protocol or via SQL*Net.
IMS	Information Management Station
AdvaInform Basic Software	The packaging of AdvaInform Basic Functions, Web support, AdvaInform User API and AdvaInform SQL*Connect.
AdvaInform Basic Objects	AdvaInform Basic Objects are pre-defined object types for storage of AdvaInform data. They can be logged by History and can be used in AdvaInform applications. They include: Master Process objects such as: DI, DO — Boolean objects (On/Off) AI, AO — Floating point objects TXT — String objects DAT — Boolean, Integer, or Floating point objects Application objects such as: coExecutor — Object that executes any program or script in the AdvaInform environment coRTM — Run Time Measurement (Master only) coBasicCalc — Object that performs a number of different calculations on 2 to 10 input values. Examples on calculations: sum, average. (Master only) MOD 300 process objects such as: DATA_FCM — Floating point objects VA_STRING_FCM — String objects You need the AdvaInform Object Handling package to use Basic Objects. This package contains a configurator to instantiate the Basic Objects. It also execute the objects during runtime.
IMS Menu	The menu that provides access to AdvaInform Basic Functions as well as optional functions.

Term	Description
Objects	In AdvaInform software (as well as other Advant technology software) the combination of data and associated procedures (operations that can be applied to the data) are represented by Objects. Objects represent significant elements or functions in the process control/process automation domain. Applications are created by combining these objects. Examples of objects are: • Process objects such as CCF loops (MOD 300) or PIDCON (Master) • History logs • Alarm and event lists • AdvaInform basic objects • AdvaInform user objects. Objects protect some of their data for the object's internal use only. Other object data, called attributes, are public and available to other objects or software applications. For example, attributes of process loop objects include the measured value, setpoint value, and base gain. Objects can read attributes in other objects via subscriptions. These subscriptions can read data from the source objects either cyclically, on demand, or when a predefined event occurs.
OCS	Open Control System
Off-Line Configuration	Creation of configuration data outside the application for later installation.
On-Line Configuration	The internal configuration of an application is directly affected while the application is active.
OS	Abbreviation for Advant Station 500 Series Operator Station.

Term	Description
Plant Network (Applicable for Master systems)	A plant wide network that can collect information from different Control Networks and also connect different Control Networks for information exchange. The plant network is a multi-protocol type of network, that is several different protocol stacks may be run in parallel over the same physical medium. An example of a protocol stack on the plant network is TCP/IP. Examples of possible protocol stacks on the plant network are TCP/IP and MB300. Different types of interaction over the plant network use different protocol stacks: • TCP/IP - X services - SQL services - File transfer services (ftp). • MB300 - Core system services. Addresses on the plant network are in accordance with each protocol respectively (that is to say, Internet addresses on TCP/IP, <netw,node>addresses on MB300, and so on). The plant network supports communication via bridges over other types of networks, such as radio links, satellite connections, various types of public networks, and so on (e.g. MasterBus 300E).
RDBMS	Relational Data Base Management System
Remote Oracle	An Oracle database in an external node is considered to be remote to the AdvaInform relational database. The two databases can be connected to each other using SQL*Net and database links. Using database links, the remote Oracle database can see the tables in the AdvaInform relational database as a distributed part of the database.
RTA	Real Time Accelerator Board. A communication board installed in an Advant Station for communication on the control network. An ABB propriety hardware that adds Real-Time capability to an HP-UX based workstation.
Subscription	A request for public data (attributes). (Subscriptions are defined by the object providing the public data.) The receiver requests the public data using the Core System subscription.
SQL	Structured Query Language. Standard language for queries towards relational databases.

Term	Description
SQL*Net	Product from Oracle Corporation. SQL*Net for TCP/IP is included in AdvaInform Basic Functions. SQL*Net is used to connect the AdvaInform relational database to remote Oracle databases and software tools supporting Oracle. SQL*NET and ODBC options to install on PCs are defined in the price list.
User Objects	User Objects are a means to develop and integrate user applications into AdvaInform. These objects are built using the AdvaBuild Object Type Builder software option. They are instantiated and executed using the AdvaInform Object Handling software option.
User Object Type	A type of object built with the Object Type Builder. The User Object Type has a framework described by the attributes, connections, commands and events and behaves as described by Operations.

1.8 Product Overview

AdvaInform Basic Functions consists of:

- a Core System that provides communication services for the system applications.
- a relational database including a license for two concurrent database users.
- an Alarm & Event Distributor.
- a User Interface system consisting of
  - a Menu system.
  - object presentation displays.
  - a System Status display.
- system management utilities.
- a single users C-development license.
- an Advant OCS communication module (optional).

AdvaInform Basic Functions is provided as part of the AdvaInform Basic Software which consists of AdvaInform Basic Functions, Web support, AdvaInform User API and AdvaInform SQL*Connect, Figure 1-3.

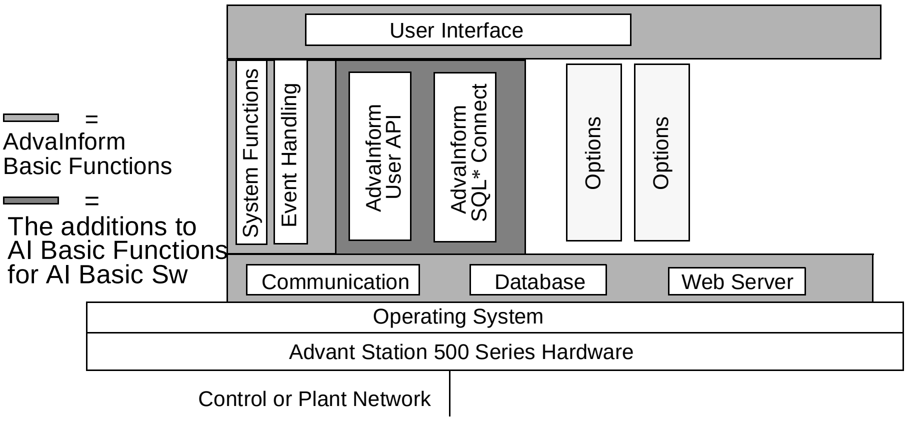


Figure 1-3. AdvaInform Basic Functions Software Structure

The additions to AdvaInform Basic Functions that the AdvaInform Basic Software provides is functionality to access the Advant OCS and to develop applications with:

- AdvaInform SQL*Connect which provides two basic functions.

  It provides the means for client applications to read data from Advant OCS objects via SQL (Structured Query Language).

  It provides the means for Advant OCS objects to read data from relational database tables on external nodes.
- AdvaInform User API (Application Programmer’s Interface) which provides a means for your application programs to interact with objects in the Advant OCS. A C-licence and C-development software is included in the Basic Functions.
- Web support - NetScape FastTrack Web server, and Netscape Navigator Gold Web browser.

1.8.1 Operating System

See the *Advant Station 500 IMS Series User’s Guide* and the HP documents listed in Table 1-1.

1.8.2 The AdvaInform Relational Database

The Advant OCS functions store data in the relational database provided in AdvaInform Basic Functions. The database is based on the Oracle RDBMS. The relational database is accessible via a wide range of tools and third-party software.

From the Advant OCS side, the relational database in AdvaInform Basic Functions is seen as any other distributed system database. This means that AdvaCommand and AdvaInform Displays are able to get data from AdvaInform Functions just as easily as they fetch data from Controllers.

The AdvaInform Basic Functions software and AdvaInform optional software use the relational database to store configuration data as well as other data. These tables are to be accessed only by applications and functions developed by ABB.

As an end user you have access to process- and application-related data in the relational database. The access is limited to views and tables related to AdvaInform applications. You do not have access to the complete Oracle RDBMS. AdvaInform Basic Functions provides the means to access these views and tables not only from the Advant Stations, but also from nodes on the Plant Network using SQL*Net.

The relational database gives the Advant OCS the features of an open system. The relational database is also a natural boundary to third-party software based on Oracle and other software that supports SQL (possibly via SQL*Net).

Further information regarding the relational database is provided in Appendix A, Relational Databases, SQL, and SQL*Plus.

Number of concurrent database users

The AdvaInform Basic Functions includes a license for two concurrent database users which is enough for many configurations. However, as soon as PC and X-terminal clients, using the database, are added to the configuration, the number of concurrent users must be extended.

You can extend the number of concurrent users to 8, 16, or 32.

1.8.3 Web Support

In order to give your Intranet users access to Advant OCS and the AdvaInform functionality, a Web Server - NetScape FastTrack - is provided.

A Web Browser - Netscape Navigator Gold is also included to provide a browser interface for all users of the IMS console, X-terminals or X-windows.

With any ordinary Web Browser you can access the Web Server to get access to the Advant OCS. Information is provided to you:

- through reports created in a viewable format with AdvaInform Reports.
- through you own applications or third party tools providing viewable information.
- through a set of predefined web pages with useful links and set-ups.

Applications and Third Party Tools

You can build your own menus, presentations and so on for the Web Server using third party tools and utilizing the Advant Station and the Netscape FastTrack Web Server as the base for the Intranet. Usage of the Web Server for your company Intranet is as extensive as you allow it to be. Through the Web your reports produced with AdvaInform Reports and any displays you have built with AdvaInform Display can be presented on any Web browser (normally executed in a PC). As AdvaInform is an open environment, you can also place other information and Web applications on top of the Web server. If your Intranet is sophisticated and has many users, the Web technique provided by AdvaInform Basic Functions can become very important for the information flow in your company.

Reports

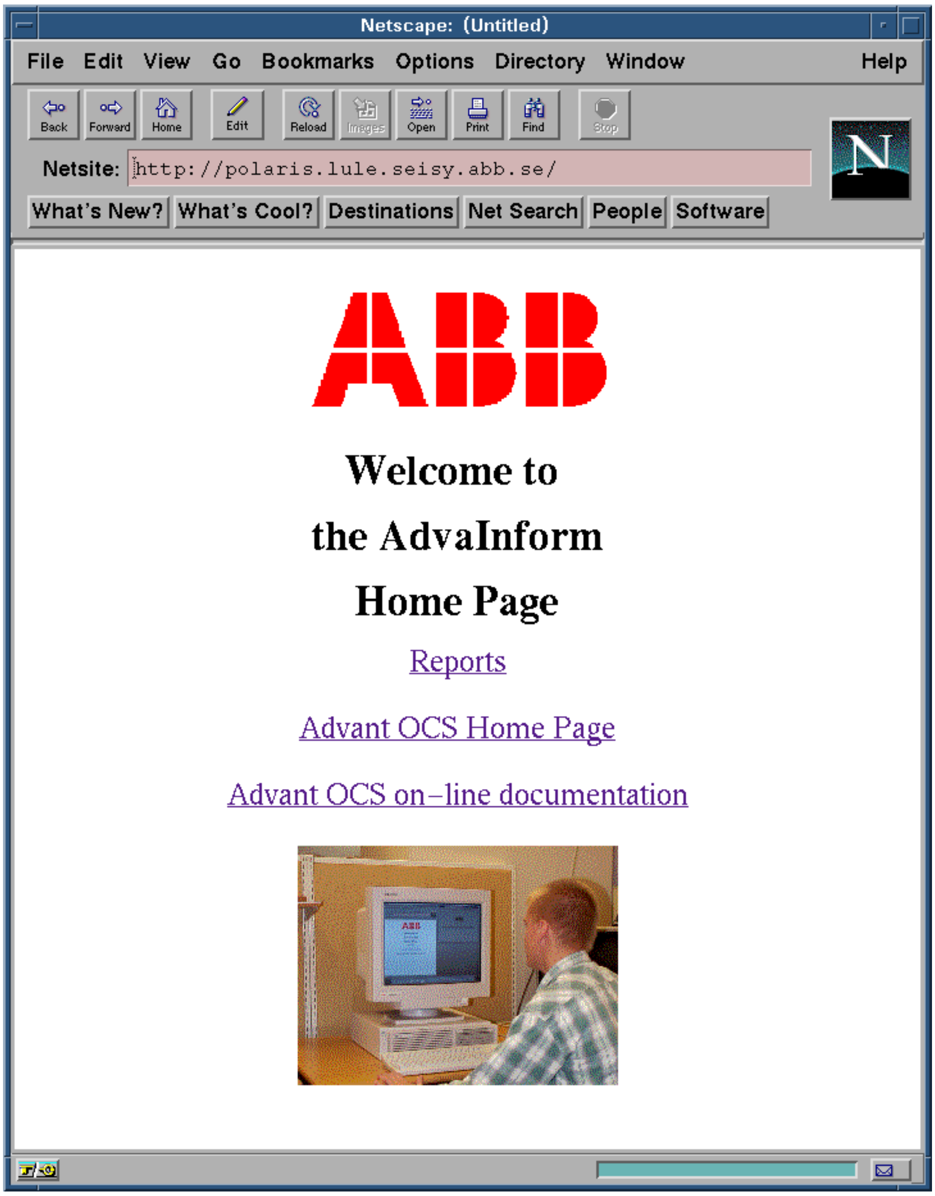
All your Reports created with AdvaInform Reports can be stored as PDF or HTML files, immediately viewable through a Web browser. The benefit of this is that the Reports need to execute only once. The resulting report is then available for everyone through an ordinary Web browser, requiring no additional software license in PC and Unix clients.

The AdvaInform Home Page

A home page is provided for your convenience, Figure 1-4. There are two links to public ABB information, available if the network has a public connection to the Internet:

- The Advant OCS web site which contains valuable information.
- The Advant OCS documentation site. A demo of the On line documentation is available. A license can be purchased from ABB which gives the right to log in and fetch the most recent versions of all on-line documentation.

The third link, useful if AdvaInform Reports is installed, contains a link to the top directory for report files.


Netscape: (Untitled)
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Edit Reload Images Open Print Find Stop
Netsite: http://polaris.lule.seisy.abb.se/
What's New? What's Cool? Destinations Net Search People Software

ABB

Welcome to
the AdvaInform
Home Page

Reports

Advant OCS Home Page

Advant OCS on–line documentation

Figure 1-4. AdvaInform Home Page

1.8.4 Communication

AdvaInform may be connected to either the Control or Plant network for Master, or to a DCN (Distributed Communication Network) or to a Plant Network for MOD 300. Communication can occur between objects that reside:

- in the same node.
- in different nodes on the same network.
- on internal (control network) nodes and objects that reside in Plant Network nodes.

Communication between objects in general, whether they reside in the same or different nodes is supported by the core system.

1.8.4.1 Advant OCS Control Network/DCN Communications

The Control Network/DCN supports communication between Advant OCS nodes, Figure 1-5. The interface between the node and the Control Network is the Real-time Accelerator (RTA) Board which is installed in the Advant Station. The control network is reserved for secure real time communication and therefore is not available for external nodes.

AdvaInform software is organized as objects which can interact either with other AdvaInform objects or with Advant OCS objects from other nodes. For example, process object values can be sent over the Control Network for storage in AdvaInform History, Figure 1-5. Values from AdvaInform objects such as Basic Objects and User Objects can also be stored.

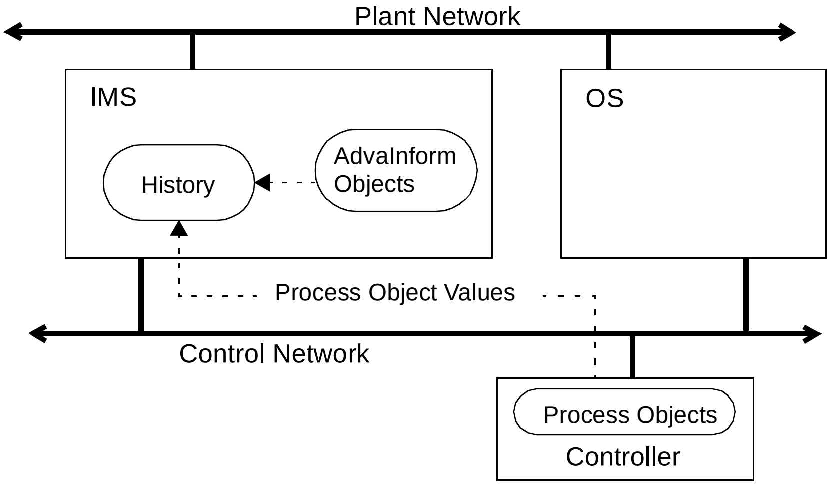


Figure 1-5. Example, AdvaInform-related Communication on the Control Network

1.8.4.2 Stand-alone Configuration

The AdvaInform Basic Functions can be installed with or without the Advant OCS communication. It can still be used provided that data from other DCS / PLC vendors is provided as object information, for example as objects in Object Handling. All the information management software can still provide most of their functionality.

If the Advant OCS communication is excluded, the RTA board is not required and the station operates as a stand-alone node, the following functions will be limited:

- The System Status functionality is limited to Node Status. Network communication and Overview Status displays (only applicable to Master), are not included.
- The Event and Alarm distribution is not supported since there is no Advant OCS to distribute to. The messages are accepted, and if the History option installed, the messages will be stored in a Message Log.

1.8.4.3 Configuration

You need to configure the Advant Station address as part of installation. See the *Advant Station 500 Series IMS User's Guide* and Section 3.6.2.1, Settings - Communication for details.

The TCP/IP address of the Advant Station must also be configured. This is done as part of the installation. See the *Advant Station 500 Series IMS User's Guide* and for details.

1.8.4.4 Plant Network Communications

The plant network consists of a IEEE 802.3 network. On the plant network you are allowed to run different protocols such as TCP/IP, MasterBus 300, GCOM, DecNet, SQL*Net etc.

The following functionality is directly supported within AdvaInform:

- TCP/IP Communication
- MasterBus 300 communication (Master Only

TCP/IP communication

This includes:

- telnet - which Is used for remote login etc.
- ftp - which Is used for file transfer
- SQL*Net communication

  SQL*Net for TCP/IP is the version of SQL*Net used by computers connected via a Plant Network (e.g. TCP/IP). SQL*Net is Oracle Corporation's network interface that allows SQL-based tools and applications running on one computer to access, manipulate, and store data in relational databases residing on other computers. There are a number of AdvaInform functions that use SQL communications with external nodes over the Plant Network. These communications are supported by TCP/IP, SQL*Net and database links.

For example:

- AdvaInform Reports can retrieve information from relational databases on external nodes, Figure 1-6. This feature allows a report to have a combination of data from the Advant OCS database and from external nodes.

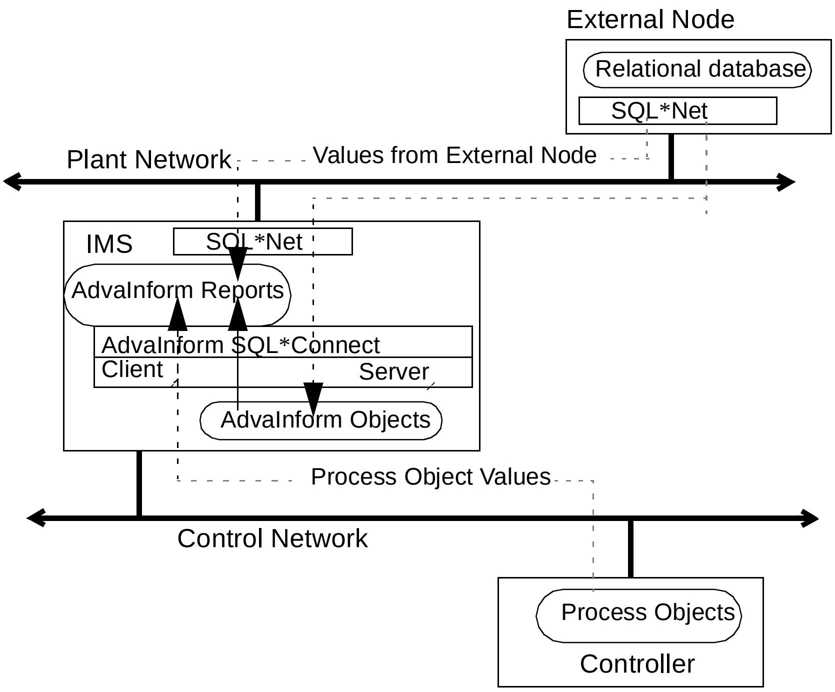


Figure 1-6. Reading Data from an External Node

- AdvaInform SQL*Connect Server provides the means for Advant OCS objects to access data that resides in relational databases on nodes on the Plant Network as if they resided within the Advant system (read functionality), Figure 1-6. For further details see the *AdvaInform SQL*Connect User´s Guide*.
- SQL-based tools and analysis packages such as spreadsheets on external nodes can read the values of Advant OCS objects via AdvaInform SQL*Connect. See Figure 1-7

– Embedded SQL in external programs can interact with AdvaInform SQL*Connect.

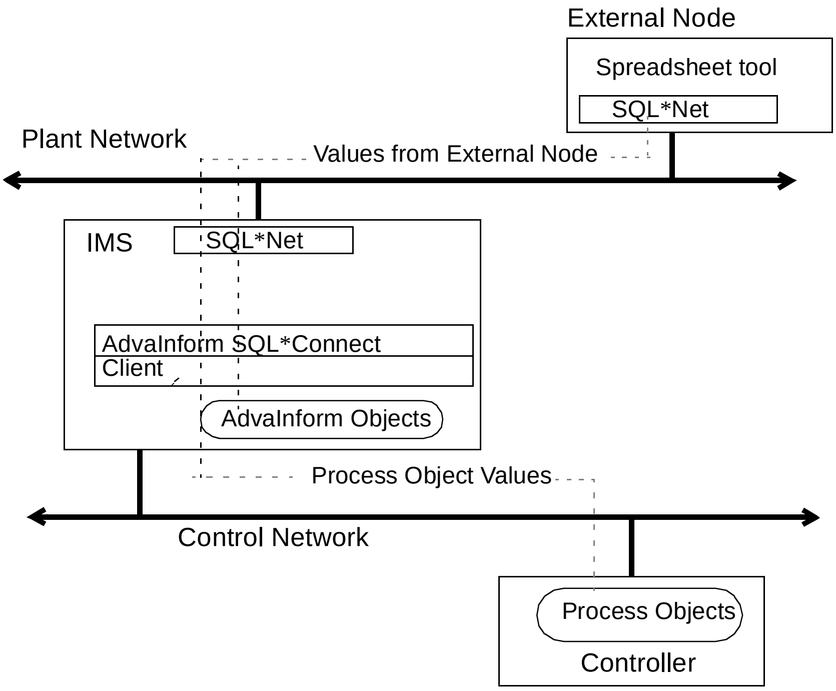


Figure 1-7. Access from an External Node

MasterBus 300 communication (Master Only)

Functions such as subscriptions, demand and alarm/event distribution operates in the same manner as within a control network.

1.8.4.5 Communication Gateways (Master)

In an Advant OCS with Master software, you can have several Control Networks and Plant Networks. They have to be connected through Gateway products. See the *MasterNet User's Guide* for details.

In IMS you have to configure which nodes are to be accessed over a Gateway. See Section 3.6.2.10, Settings - Node Broker (Master) for instructions.

1.8.5 User Interface and Menu Systems

The User Interface is based on X-technology, which makes it possible for AdvaInform to act as a server for X-clients on the TCP/IP Network. AdvaInform is a multi-user environment, so several users can simultaneously access AdvaInform functions from X-clients on the TCP/IP Network.

X-technology makes it possible for you to interact with applications that are executing in several nodes on the TCP/IP Network. From an X-terminal, you can manage AdvaInform in one terminal window, an order system in a second, and a word processor in a third.

The HP-UX user interface is based on HP Common Desktop Environment (HP CDE). For information regarding this user interface, refer to the *HP Common Desktop Environment User's Guide*.

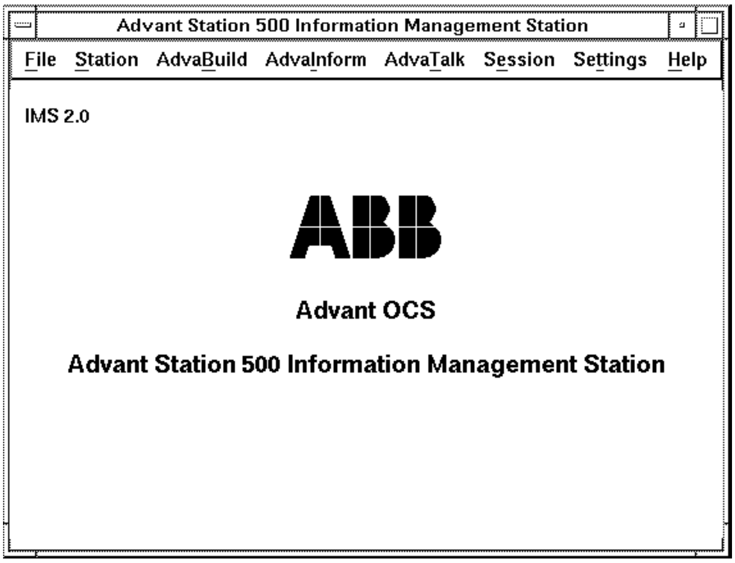
1.8.5.1 IMS Menu

Access to AdvaInform Basic Functions and optional AdvaInform, AdvaBuild and AdvaTalk functions is provided via the menu, Figure 1-8. This menu is displayed when you log into the IMS Menu as described in Section 3.3, Application Start-up and User Authorization. There are three user levels (categories) associated with the menu:

- Operator
- Engineer
- System

The scope of functions that you can access on this menu depends on the user level that you log in at. Functions that are not available to you at your current user level are dimmed.

A brief summary of the functions available via this menu is provided in Table 1-2.



IMS 2.0

Advant OCS

Advant Station 500 Information Management Station

Figure 1-8. IMS Menu

Table 1-2. AdvaInform User Interface Menu Items

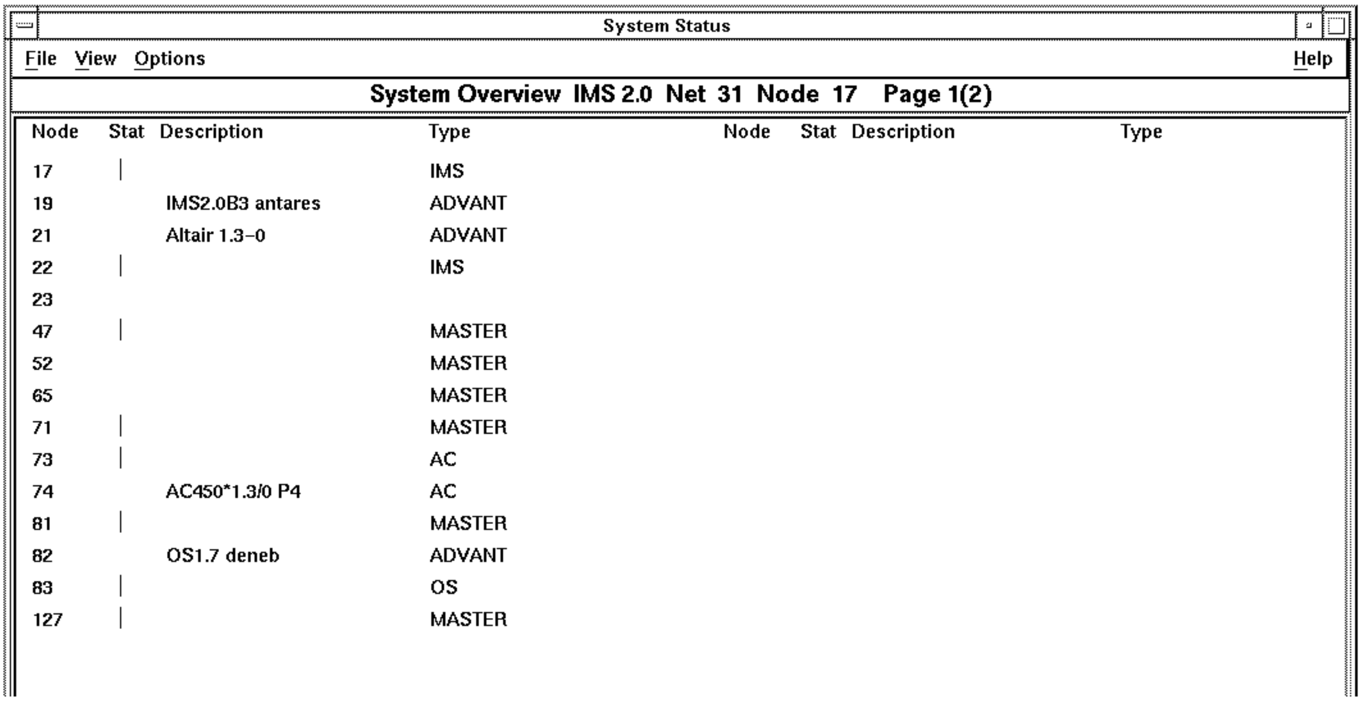
Menu	Item	Function
File	Exit	Return to where you started the menu
Station	System Status	Status displays that present status of IMS software, hardware, peripherals and connected MasterNet nodes. (full function for Master, partial function for MOD 300)
	Object Presentation	View object attributes and status information.
	IMS Terminal Window	Access to the Operating System, HP-UX.
	Database Access	SQL*Plus interface to AdvaInform relational database and external databases.
	Editor	Starts a text editor. Default editor for Advant Station 500 Series is HP-CDE text editor.
	Mail	Starts electronic mail. Default mail package for Advant Station 500 is HP-elm.
	System Messages	Displays system message.
	Object Reference Maintenance	Object reference maintenance display. (Master only)
AdvaBuild	Object Type Builder	Access to AdvaBuild Object Type Builder.
	Central Backup	Backup of connected Controllers, MasterGates and MasterView 800/1. (Master only)
	Programmer 2000 - Pro*C	Terminal window which gives you access to the Programmer 2000 - Pro*C pre-compiler.
AdvaInform	Object Handling	Submenu access the Object Configuration and Object Handlers.
	SQL Connect	Submenus access either the Client configurator or the Server configurator.
	User API	Access to the User API examples and the C-compiler.
	History (including PDL)	Submenu access to the TTD-configuration or History Builder.
	Calculations	Submenus access either the Calculation Builder or the Calculation Status. (Master only)
	Reports	Submenus access Report Configuration, Manual Execution, Report Status and View Logs.
	Event Log	Submenus access Event Log Configuration and Event Log Status. (Master only)
	SPC	Submenu access to SPC. (MOD 300 only)

Table 1-2. AdvaInform User Interface Menu Items (Continued)

Menu	Item	Function
AdvaTalk	Link to MATLAB	Submenu access to the Link to MATLAB configuration.
Session	Show User	Displays current user and user's category.
	Language	Submenus access to US English, German and Swedish.
	Change Password	Change menu password of current user.
Settings	Communication	Defines communication parameters for network and node.
	User Definition	Create, modify and delete users.
	Station Backup	Makes a backup of the Advant Station.
	Process Supervision	Defines process start up and supervision.
	Maintenance	Starts SAM, an HP standard tool for system maintenance.
	System Variables	Defines default system variables for the menu. Generally used to change the Editor and Mail System.
	System Messages	Set up the size and limits for the System messages Display.
	RTA	Access to RTA terminal window. Allows access to RTA board and to connected Controllers. (Master only)
	Date and Time	Used to make IMS clock master to synchronize Advant OCS nodes.
	Node Broker	Define nodes outside the control network. (Master only)
Help	Several entries	Help information.

1.8.5.2 System Status

The System Status is an important means of supervising and presenting the state of the Advant Station as well as the installed AdvaInform, AdvaBuild, AdvaTalk and user application software, Figure 1-9.



System Overview IMS 2.0 Net 31 Node 17 Page 1(2)

Node	Stat	Description	Type	Node	Stat	Description	Type
17	│		IMS				
19		IMS2.0B3 antares	ADVANT				
21		Altair 1.3-0	ADVANT				
22	│		IMS				
23							
47	│		MASTER				
52			MASTER				
65			MASTER				
71	│		MASTER				
73	│		AC				
74		AC450*1.3/0 P4	AC				
81	│		MASTER				
82		OS1.7 deneb	ADVANT				
83	│		OS				
127	│		MASTER				

Figure 1-9. System Status display

System Status Presentation

You present the system status through status displays activated by choosing **System Status** from the **Station** menu, Table 1-2.

The following function is common for Master and MOD 300:

- you use the System Status display to show the status of all connected hardware devices and currently installed software functions (AdvaInform, AdvaBuild, AdvaTalk as well as third party or user applications). See Figure 1-9.

The following functions are Master specific:

- you use the Overview display to get an overview of the functional status of all Advant Master nodes connected to the MasterBus 300 network.
- you use the Network Status display to get an overview of the communication status of all Advant Master nodes connected to the MasterBus 300 network.

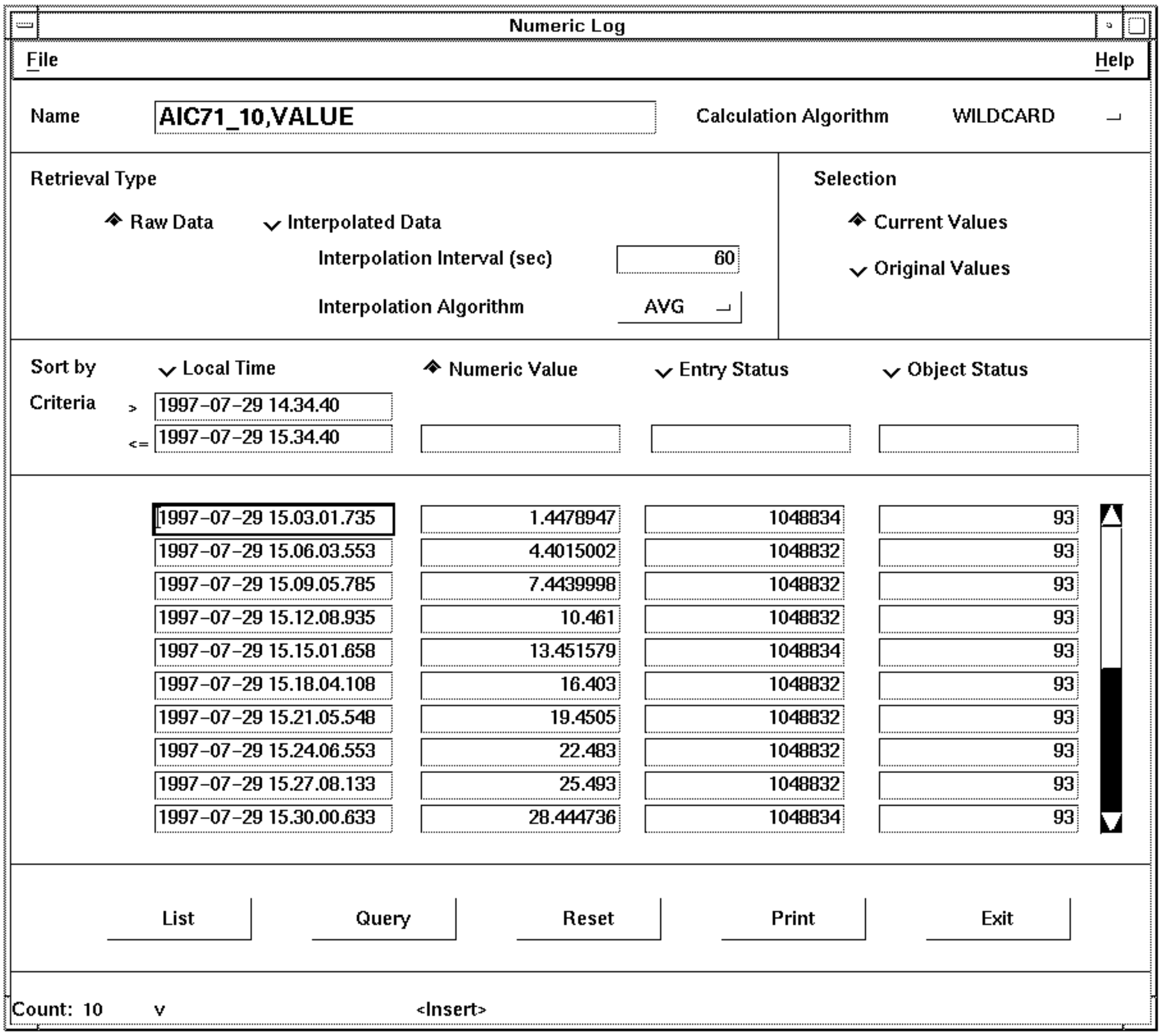
Operation and presentation of system status displays are described in Chapter 4, Runtime Operation.

Software Supervision Configuration

You can configure the execution and supervision of applications using the Process supervision form. See Section 3.6.2.4, Settings - Process Supervision.

1.8.5.3 Object Presentation Access

Various Object Presentation forms are available for displaying information from objects and logs Figure 1-10.



Name AIC71_10,VALUE

Calculation Algorithm WILDCARD

Retrieval Type

● Raw Data ○ Interpolated Data

Interpolation Interval (sec) 60

Interpolation Algorithm AVG

Selection

● Current Values

○ Original Values

Sort by ○ Local Time ● Numeric Value ○ Entry Status ○ Object Status

Criteria > 1997-07-29 14.34.40

<= 1997-07-29 15.34.40

1997-07-29 15.03.01.735	1.4478947	1048834	93
1997-07-29 15.06.03.553	4.4015002	1048832	93
1997-07-29 15.09.05.785	7.4439998	1048832	93
1997-07-29 15.12.08.935	10.461	1048832	93
1997-07-29 15.15.01.658	13.451579	1048834	93
1997-07-29 15.18.04.108	16.403	1048832	93
1997-07-29 15.21.05.548	19.4505	1048832	93
1997-07-29 15.24.06.553	22.483	1048832	93
1997-07-29 15.27.08.133	25.493	1048832	93
1997-07-29 15.30.00.633	28.444736	1048834	93

Count: 10 v <Insert>

Figure 1-10. Example of the Object Presentation Form

1.8.5.4 Terminal Window

You can open a terminal window via the **Terminal Window** menu item in the **Station** menu. When you open a terminal window this way, rather than via the terminal window icon, the proper AdvaInform environment will automatically be set up.

1.8.5.5 Mail

You can access an electronic mail package via the **Mail** menu item in the **Station** menu. The default mail package for the Advant Station is HP-elm which is included with HP-UX. You can change the mail package via the **Define System Variables** menu item in the **Settings** menu. See Section 3.6.2.6, Settings - System Variables.

1.8.5.6 System Messages

You can use the System Message display to verify that there are no system messages from the installed AdvaInform software. Application programs can create their own system messages, available in that display, via the AdvaInform User API.

The System Message display has features to selectively display the current logs in the system and also to control presentation in other ways.

1.8.5.7 Editor

You can access a text editor via the **Editor** menu item in the **Station** menu. The default editor for the Advant Station is the HP CDE text editor. You can change the editor via the **Define System Variables** menu item in the **Settings** menu. This procedure is described in the *Advant Station 500 Series IMS User´s Guide*.

1.8.6 Alarm & Event Handling

The Alarm and Event Support in the AdvaInform Basic Functions differs for Master and MOD 300:

Master

AdvaInform Basic Functions support Alarm and Event Handling, Figure 1-11. All events generated by applications residing in the IMS can be stored in a History message log and be distributed to event subscribers. This means that all events generated by AdvaInform objects will be distributed to operator stations and MasterViews resident in the same control or plant network. Alarm acknowledge is available via those operator stations.

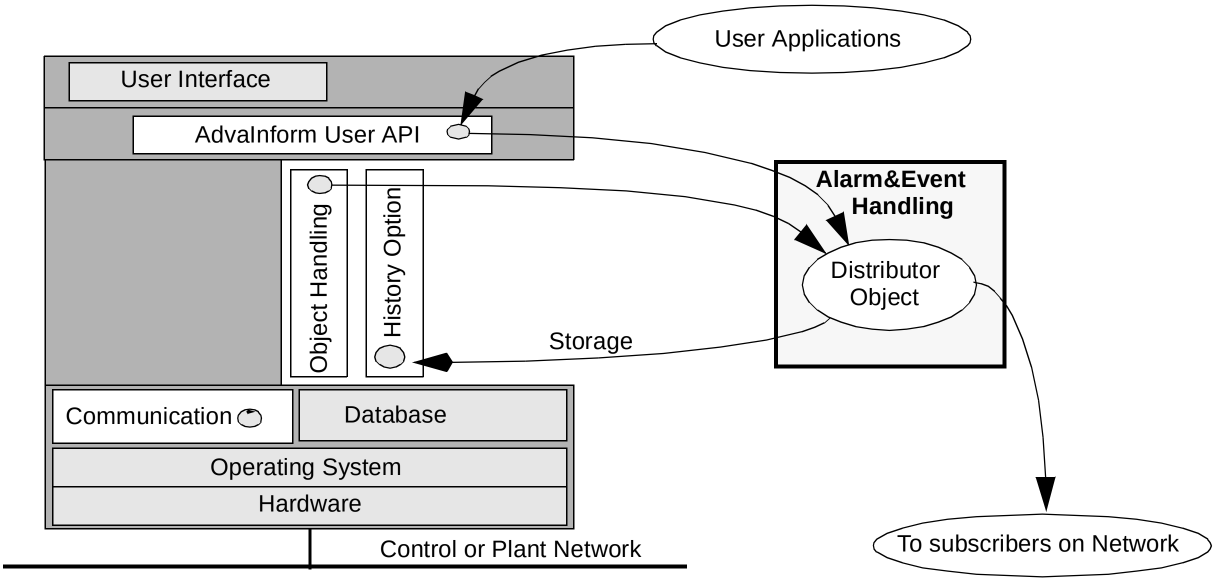


Figure 1-11. Alarm and Event Storage and Presentation

All event messages stored in a Message Log can be viewed using the Message Log object form. You can develop applications as User Objects with the AdvaBuild Object Type Builder option. Support for the generation of events and the interaction with alarm lists is provided in User Objects.

The Basic Objects can generate both alarms and events. Applications that interact with Basic Objects can use this to generate alarms and events. As an example, you can use Basic Objects from Object Access Services to indirectly generate alarms and events.

You can also generate events directly via AdvaInform User API. The events are sent as operations to the event and alarm distributor object.

MOD 300

AdvaInform Basic Functions support distribution of TCL Billboard messages to MOD 300 operator stations, Figure 1-12.

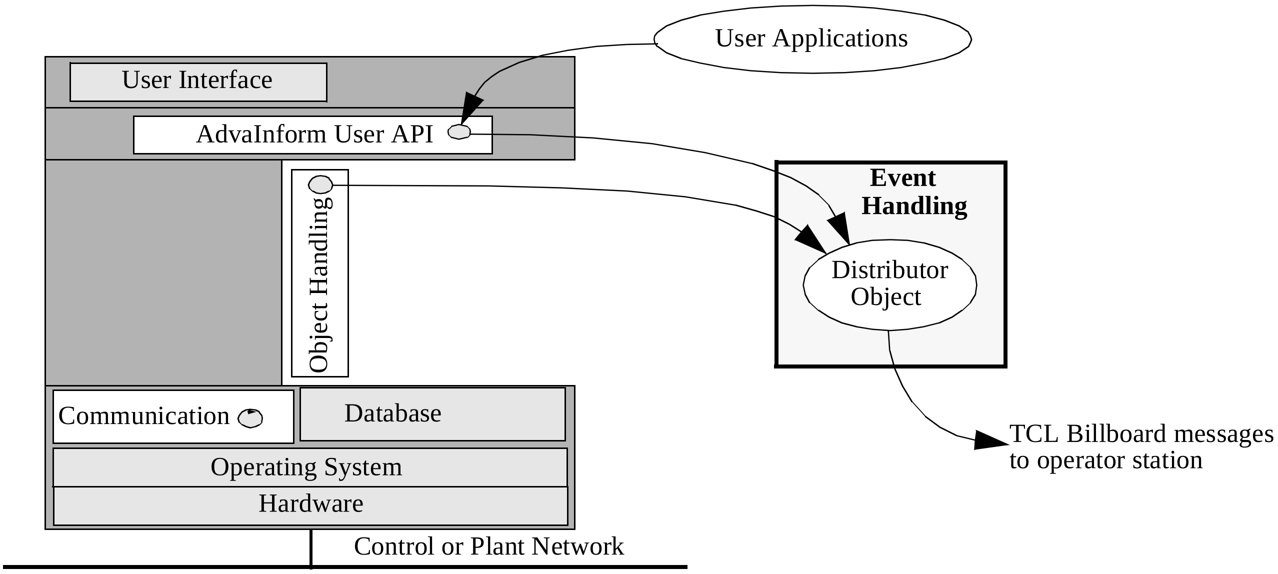


Figure 1-12. Event Handling

1.8.7 System Administration

The following sections provide a brief overview of IMS administration functions. For detailed instructions, refer to the *Advant Station 500 Series IMS User´s Guide*.

1.8.7.1 User Definition

User Definition provides the means to create, modify and delete AdvaInform Users. When you create a user you can define the default AdvaInform category, password and language for the user. As a result you will also get an AdvaInform user, an HP-UX user and a corresponding Oracle user. For details see Section 3.6.2.2, Settings - User Definition and the *Advant Station 500 Series IMS User´s Guide*.

1.8.7.2 Daylight Savings and Time Synchronization

Daylight Saving

The Daylight Savings functionality of all Advant Stations is configured as part of the installation procedure. As most Advant Stations are pre-installed when they are delivered, the Daylight Savings functionality and the related Time Zone selection is already done. See the *Advant Station 500 Series IMS User´s Guide* for details on how to do the installation, if necessary, and how to check that the Daylight Savings and Time Zone are properly configured.

NOTE

The Daylight Savings and Time Zone set-up must be identical for all Advant Stations in your system. If not, History logging, Trend presentation and other time related functions may be severely disturbed when Daylight Savings time shifts occur. Check this as soon as you add any new Advant Station to your network.

Time Synchronization

The Advant OCS method for time synchronization is described in *Advant Station 500 Series IMS User´s Guide*. Consult that manual to see how your Advant Station can synchronize or be synchronized to a common system time.

NOTE

Time may not be set in any other way than by the **Settings - Date & Time** dialog since this will affect the system in an unpredictable way.

1.8.7.3 Station Backup

Backup and restore functions are provided via the **Station Backup** menu item in the **Settings** menu. The backup function provides a full and incremental backup of the AdvaInform Basic Functions as well as installed software options. You can also start History Backup from the station Backup function.

Station Backup is required during installation and upgrade. Details on Station Backup are provided in the *Advant Station 500 Series IMS User´s Guide*.

1.8.8 AdvaInform Application Development Tools

The application development tools provided with AdvaInform Basic Software, Figure 1-3, are the AdvaInform User API (Application Programming Interface) and the SQL services including AdvaInform SQL*Connect. There are also additional development tools provided by the AdvaBuild Object Type Builder, Programmer 2000-Pro*C and AdvaInform SQL*Connect Programming options.

AdvaInform Basic Functions provides application portability between different Advant Station platforms and system generations. New applications developed on the platform have the same access to system functions as system applications, thus giving full but controlled access to system services.

The main function of the AdvaInform development tools is to provide access to the process database and AdvaInform relational database through a third generation programming language (3-GL) and SQL interface. You can use the development tools to develop applications in C and C++ - a 3-GL language. These programs contain functions which read from or write to, or otherwise manipulate Advant OCS objects.

Other applications can be built with third-party tools such as Oracle Forms[1]. A benefit of the AdvaInform client - server approach to computing is that these tools may be installed and operated from other nodes.

C and C++ developers licenses

The AdvaInform Basic Functions includes a license for C-development for one user, normally used when developing User API programs. If several users need to work with C development, licenses for more users are available as an option.

As an alternative, C++ can be used for the development of User API programs. A C++ single users development environment is available as an option. If several developers will use it, two or more licenses can be purchased.

1.8.9 Support for Optional Software

This includes AdvaInform, AdvaBuild, AdvaCommand and AdvaTalk options.

See *Advant Station 500 Series IMS User's Guide* for more information.

---

[1] User's licenses for Oracle Forms are not included in AdvaInform Basic Functions

Chapter 2 Installation

2.1 Site Planning and Environment

For information regarding site planning and environmental considerations for your IMS, refer to the relevant *Advant Station 500 Series Hardware User's Guide*.

2.2 Setup, Start up and Shut down

For instructions regarding hardware setup and installation for the IMS, refer to the relevant *Advant Station 500 Series Hardware User's Guide.* Also refer to the *Advant Station 500 Series IMS User's Guide* for information regarding connection of peripheral devices to start up and shut down the AdvaInform software (and IMS).

Chapter 3 Configuration/Application Building

3.1 Design Considerations

This chapter describes how to use AdvaInform Basic Functions to configure information management functions in your system. The following topics are covered:

- AdvaInform User Authorization
- How to log in to the IMS Menu and/or AdvaInform relational database

  When you log in you should consider what functions you need access to. You can log in to the IMS Menu at one of three user levels: operator, engineer, or system. The level you log in at determines what functions on the IMS menu you have access to. Also, rather than log in to the IMS Menu, you can log in to the AdvaInform relational database. Refer to Section 3.3, Application Start-up and User Authorization for details.
- How to open the IMS menu from an Operator Station:
  - Section 3.5.2.2, Master Operator Station.
  - Section 3.5.2.3, MOD 300 Operator Station.

3.2 Capacity and Performance

Information over capacity and performance are available in the *Advant Station 500 Series IMS User's Guide*.

3.3 Application Start-up and User Authorization

3.3.1 User Authorization

The username and password that you use to log in to HP-UX determines the scope of your privileges within AdvaInform.

User authority is established by assigning a user to one of the categories listed in Table 3-1 when the user is created. Creating a number of IMS users is one of the first tasks the system manager should do after the first installation. See the *Advant Station 500 Series IMS User´s Guide* for details.

Table 3-1. User Categories

Category	Oracle Role	Privileges
SYSTEM	IMS_SYSTEM_ROLE	Define, Configure, Operate, Opsyst
ENGINEER	IMS_ENGINEER_ROLE	Configure, Operate, Opsyst
OPERATOR	IMS_OPERATOR_ROLE	Operate

The privileges assigned to IMS menu users are described in Table 3-1. In Table 3-3 the required user categories and access privileges are listed for the different items in the Menu

Table 3-2. User Privileges

Privilege	**Description**
Define	System administration functions such as setting the system time, setting up node numbers, and so on.
Configure	Configuration of object instances, history logs, report templates, and so on.
Operate	Operation of the IMS runtime functions such as object forms.
Opsyst	Provides access to the operating system through the terminal window accessed via the IMS menu.

.

Table 3-3. Privileges Required for IMS Menu Items

Menu	**Menu Item**	**Required Privilege**
File	Exit	operate
Station	System Status	operate
	Object Presentation	operate
	IMS Terminal Window	opsyst
	Database Access	operate
	Editor	operate
	Mail	operate
	System Messages	operate
	Object Reference Maintenance	operate
AdvaBuild	Object Type Builder	configure
	Central Backup	configure
	Programmer 2000-Pro*C	configure

Table 3-3. Privileges Required for IMS Menu Items (Continued)

Menu	Menu Item	Required Privilege
AdvaInform	Object Handling	operate
	Object Configuration	configure
	Object Handlers	define
	Object Status	operate
	SQL*Connect	configure
	Client - object instances	configure
	Client - object types	configure
	Server Configuration	configure
	User API	configure
	History	operate
	History Configuration	configure
	TTD-Configuration	configure
	History Status	operate
	Calculations	operate
	Calculation Builder	configure
	Calculation Status	operate
	Reports	operate
	Reports Configuration	configure
	Manual Execution	operate
	Report Status	operate
	View Logs	operate
	SPC	operate
	SPC Configuration	configure
	Event Log	operate
	Event Log Configuration	configure
	Event Log Status	operate
AdvaTalk	Link to MATLAB	operate
	MATLAB	configure
	Configuration	configure

Table 3-3. Privileges Required for IMS Menu Items (Continued)

Menu	Menu Item	Required Privilege
Session	Show User	operate
	Language	operate
	US English	operate
	German	operate
	Swedish	operate
	Change Password	operate
Settings	Communication	define
	User Definition	define
	Station Backup	define (plus root password)
	Process Supervision	define
	Maintenance	define (plus root password)
	System Variables	define
	System Messages	define
	RTA	(access to ABB internal only)
	Date and Time	define
	Node Broker	define
Help	all	operate

3.3.2 HP-UX Users

The following HP-UX users are predefined within AdvaInform Basic Functions:

- oracle: Owner of the Oracle database.
- root/superuser: A password is needed to use the HP tool SAM for maintenance of the AdvaInform software.
- ocsmgr: manager. Used when logging onto the IMS Menu as ocsmgr. It is recommended that the predefined password for ocsmgr be changed the first time you open the IMS menu.

Your home directory and environment are set up in the same way for all users belonging to the same category.

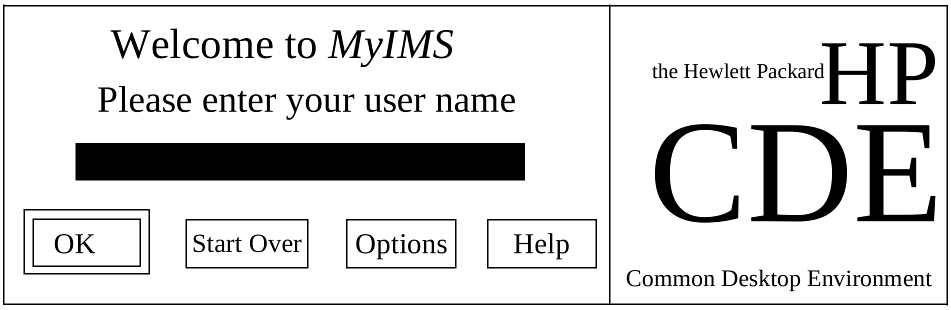
- Your home directory is under `/home/`*`UserName/`*
- The AdvaInform environment is automatically set-up.

3.3.3 Log In Procedure

To log into the IMS Menu:

1. First log in to HP-UX at the appropriate user-level.

   When you power up, the HP CDE Login Screen with the first log in dialog box is displayed, showing user name of the station, Figure 3-1.

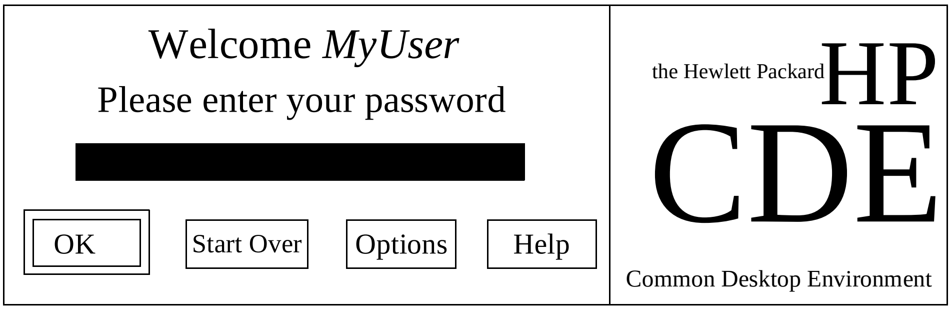


Welcome to *MyIMS*

Please enter your user name

Figure 3-1. HP-UX First Log in Dialog Box

   a. Enter the user name in the field, then click **OK** (or press <**Return**>). This displays a second dialog box that shows the user name you entered, Figure 3-2.



Welcome *MyUser*

Please enter your password

Figure 3-2. HP-UX Second Log in Dialog Box

   b. Enter the password and click **OK** (or press <**Return**>). This opens the HP CDE environment. A terminal window may already be open.

      If the IMS menu start-up command is included in the system log-in file for a user, the IMS Menu will be opened automatically when you log in to HP-UX.

      If this start-up command is not included in the system log-in file, continue with step 2 to log into the IMS Menu.

2. Open the IMS menu.

   There are two methods for opening the IMS menu. You can either click on the IMS menu icon on the bottom of the HP CDE front panel, Figure 3-3, or you can enter the IMSmenu command via a terminal window command line.

   Method 1 - IMS Menu Icon

   The IMS menu icon can be used when you are logged in at either the operator or engineer category level. To log in using the IMS menu icon, click on the IMS menu icon in the bottom of the HP CDE Front Panel, Figure 3-3.

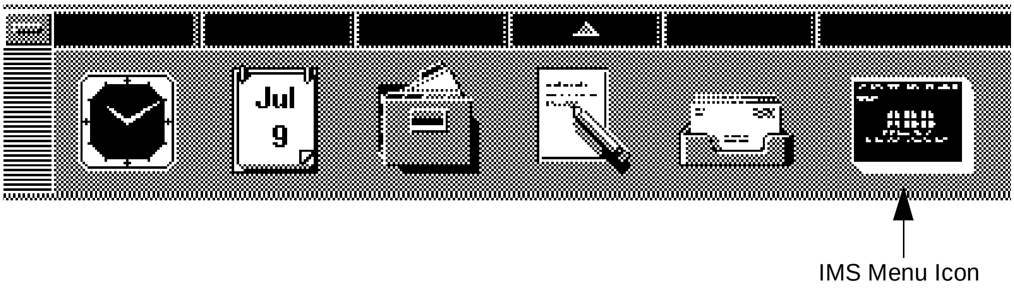
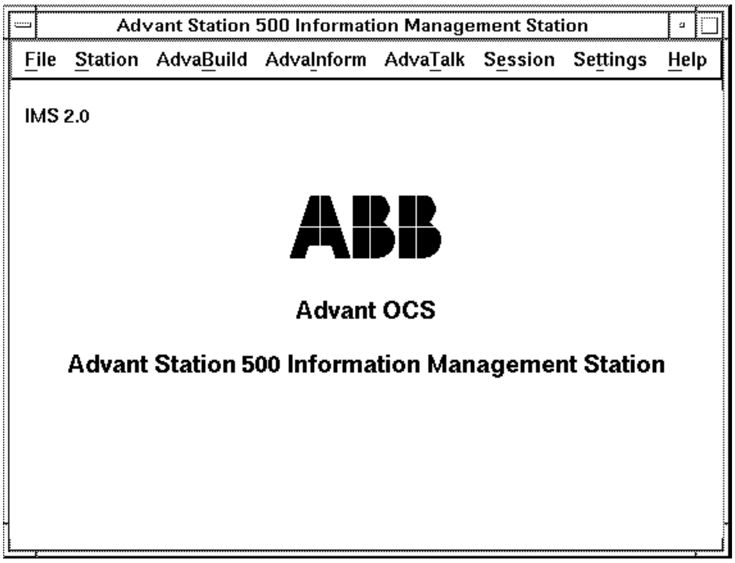


Figure 3-3. IMS Menu Icon

   This opens the IMS menu, Figure 3-4,with the same user that is logged in to HP-UX (in step 1).


Advant Station 500 Information Management Station
File Station AdvaBuild AdvaInform AdvaTalk Session Settings Help

IMS 2.0

Advant OCS

Advant Station 500 Information Management Station

Figure 3-4. IMS Menu Window

Method 2 - Command Line

As an alternative to the IMS menu icon, you can use the following procedure:

a. Open a terminal window by clicking on the terminal window icon.

b. Enter: **`IMSmenu`** at the `$` prompt, then press **<Return>**. This opens the IMS menu, Figure 3-4,with the same user that is logged in to HP-UX (in step 1).

3.3.4 Log Off Procedure

To log off the Advant Station:

1. Close all AdvaInform applications to avoid problem at next start-up.

NOTE

This is important because at log in, CDE automatically starts all applications that were running last time a user logged off. The environment will then be incorrect for all automatically started AdvaInform applications. This may cause unpredictable behavior, incorrect colors, and so on.

2. Click on the **EXIT** button at the left bottom corner of the HP CDE front panel, Figure 3-5.

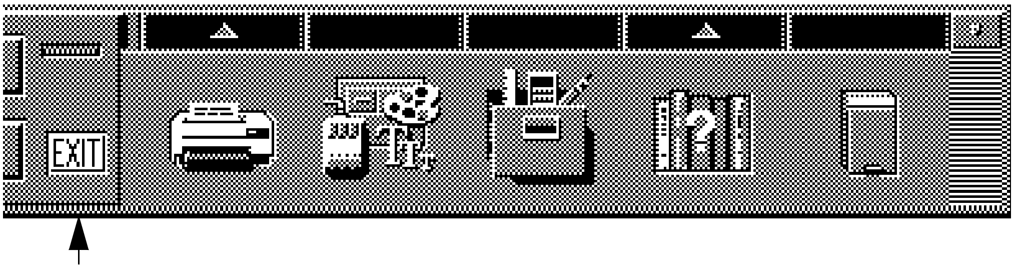


Figure 3-5. EXIT Button

A pop up dialog will then ask whether or not you want to continue the log off. If so, confirm. This completes the log off.

3.3.5 Log In to the AdvaInform Relational Database

All users have an Oracle account with the same name as the IMS user, plus the prefix `ops$`. For example the IMS user *testop* has an Oracle account `ops$testop`. An Oracle OPS account automatically uses the operative system user name and password when connecting to Oracle, therefore no user name/password has to be given.

If you are already logged into the IMS Menu, you can log in to the AdvaInform relational database using the **Database access** menu item in the **Station** menu. Or, if you are not logged into the IMS Menu, you can use a terminal window to start sqlplus (Automatically uses your oracle user *ops`$MyUserName`*):

```
$ sqlplus /
```

External login over SQL*NET to such a user is done as follows:

```
$ sqlplus ops\$testop/<password>@<service_name>
```

where:

password is the password you defined with user definition

service name is an entry in the file /etc/tnsnames.ora. To log on the AdvaInform database, service_name is normally the same as host name.

3.3.6 Oracle users

The Oracle user belongs to a certain role depending on the IMS user category. See Table 3-1 and Table 3-4..

Table 3-4. Oracle roles

IMS Category	Oracle Role	Privileges
SYSTEM	IMS_SYSTEM_ROLE	IMS_ENGINEER_ROLE and CREATE USER ALTER USER DROP USER
ENGINEER	IMS_ENGINEER_ROLE	IMS_OPERATOR_ROLE and CREATE DATABASE LINK CREATE SYNONYM CREATE VIEW
OPERATOR	IMS_OPERATOR_ROLE	CREATE SESSION ALTER SESSION

All Oracle Roles have the following privileges:

- Read Process Objects and System Application Objects via the SQL*Connect Client.These objects include FCM objects, CCF continuous and device loop objects, AdvaInform Basic Objects, TCL objects, and historical log objects
- Read the CO_xx_INST tables, owned by the USEROBJECTS user, for example:

  ```
  SELECT * FROM USEROBJECTS.CO_AI_INST;
  ```
- Insert, update, and delete the DB_xx tables.

As IMS_ENGINEER_ROLE or higher you can also

- Create Database Links, Synonyms and Views.

3.3.7 IMS Menu Icon

The IMS menu icon serves as an indicator that shows whether or not the AdvaInform is running. If the IMS menu icon is filled all necessary processes are up and running. If the icon is transparent as shown in Figure 3-6, the AdvaInform is either stopped or starting. In both cases, the IMS menu can be activated via the icon.

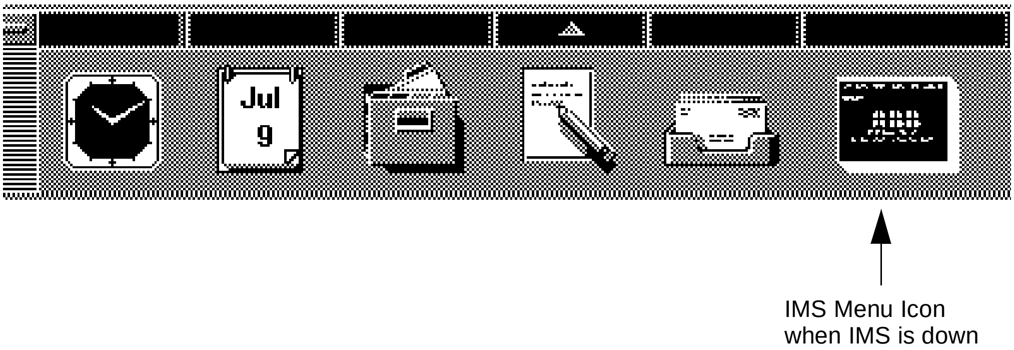


Figure 3-6. IMS Menu Icon When AdvaInform is Stopped or Starting

3.4 Tutorial

Tutorials for applications that are accessible via the IMS menu are provided in the respective user's guides. For instance, tutorials for using SQL*Connect are provided in the *AdvaInform SQL*Connect User's Guide*. Tutorials for configuring the History data base are provided in the *AdvaInform History User's Guide*, and so on.

3.5 Application Procedures

3.5.1 Application Building within AdvaInform Basic Unit

AdvaInform Basic Functions does not include any application development tools. The application development tools are included in the AdvaInform Basic Unit. The application development tools provided with the AdvaInform Basic Unit are the AdvaInform User API (Application Programming Interface) and the SQL services including the AdvaInform SQL*Connect. Detailed instructions for using these development tools are provided in their respective users guides. Refer to Section 1.5, Related Documentation.

3.5.2 Access to AdvaInform Functionality from an Operator Station

3.5.2.1 Available Displays

The AdvaInform functions that are available from the IMS menu are system variables as described in Table 3-5. When you log in as ocsmgr you can choose **Define System Variables** from the **Settings** menu on the IMSmenu to see the definitions of the respective variables. Then use that definition in your remote log-in, your application menu or whatever the use may be.

NOTE

Make sure that the authorization of the respective application is respected. For example, ordinary Operator Station operators should not be allowed to reconfigure a History log.

Table 3-5. Useful System Variables

Variable name (Token)	Description	Category
OBJFORMS_AI	AI Object Forms Display	Operator
OBJFORMS_AO	AO Object Forms Display	Operator
OBJFORMS_DAT	DAT Object Forms Display	Operator
OBJFORMS_DI	DI Object Forms Display	Operator
OBJFORMS_DO	DO Object Forms Display	Operator
OBJFORMS_MANSTN	MANSTATION Object Forms Display	Operator
OBJFORMS_PIDCON	PIDCON Object Forms Display	Operator
OBJFORMS_RATIOSTN	RATIOSTATION Object Forms Display	Operator
OBJFORMS_TEXT	TEXT Object Forms Display	Operator

Table 3-5. Useful System Variables (Continued)

Variable name (Token)	**Description**	**Category**
OBJFORMS_NUMERIC_LOG	Numerical Log Display	Operator
OBJFORMS_MESSAGE_LOG	MESSAGE LOG Display	Operator
EVENT_LOG_PRESENTATION	Event Log presentation display	Operator
SYSTEM_MAIL	System Mail	Depends on login
DATABASE_ACCESS	SQL*PLUS	Depends on login
SYSTEM_STATUS	System Status Display	Operator
TERMINAL	Open an IMS Terminal	Operator
HISTORY_STATUS	History run time status display	Operator
OBJECT_STATUS	Object Status Display	Operator
CALCULATION_CONTROL	Calculation status display	Operator
EVENT_LOG_STATUS	Event Log status display	Operator
REPORT_STATUS	Report Status display	Operator
REPORT_MANUAL	Manual execution of Report	Operator
OBJECT_TYPE_BUILD	Start a terminal window for usage of Object Type Builder utilities	Engineer
OBJECT_CONFIGURE	Object Configuration display	Engineer
OBJECT_HANDLERS	Object Handlers terminal window	Engineer
HISTORY_CONFIGURE	History configuration	Engineer
CALCULATION_CONFIGURE	Calculation builder	Engineer
REPORT_CONFIG	Report configuration display	Engineer
EVENT_LOG_CONFIGURE	Event Log configuration display	Engineer

3.5.2.2 Master Operator Station

Process Displays in the operator station can be configured to allow log-on to AdvaInform by dynamic function keys. The following describes how this can be done. An equivalent configuration is needed if another workstation is to access the IMS Menu.

1. Make sure that the TCP/IP address of the AdvaCommand node is known to the AdvaInform node and vice versa. This is normally done when the Advant Station is configured. You can use the */etc/ping* tool to verify that the OS is known to the IMS.

   You should include the name and TCP/IP address of the other node in the */etc/hosts* file for each of the nodes (IMS and OS).

2. Configure the OS to allow the IMS node to act as X host. For example, enter the following command from an OS terminal window:

   ```
   xhost + <IMS TCP/IP address>
   ```

3. Create a new user on the IMS with the appropriate privileges to execute the AdvaInform functions. See Section 3.5.2.1, Available Displays for a list of privileges for each function.

4. Copy the shell script **/users/imsext/IMSextern** to the home directory of the new user. Executing this script sets up the environment. By default it also starts the IMS menu. To start something else, see Section 3.5.2.1, Available Displays for instructions how to find the start-up string for other AdvaInform functions.

   The remote host (OS) must be defined in IMS **/home/.rhosts**, (see **man hosts.equiv**). For example:

   ```
   as1 customer
   as2 customer
   ...
   ```

   That means the name of the OS node shall be in the .rhosts file followed by "customer". You can get the name of the OS node by entering: `hostname` in an OS terminal window.

5. If a workstation other than the OS is used, open a terminal window and execute:

   ```
   remsh <IMS station name or IP address> -l imsext -n IMSextern -display <workstation name or IP address>:0.0
   ```

   If the IMS menu is to be accessed from an OS, carry out the following configuration:

   a. Start up AdvaBuild Display Builder.

   b. Choose the process display you want to add a dynamic function key to.

   c. Invoke configuration of display attributes.

   d. Configure a dynamic function key as for an external application, as shown below:

      Function: IMSextern
      Type: Remote
      User name: imsext
      Host name: AdvaInform node name or IP address

   e. Install the process display

f. Choose the display from the process display menu and push the dynamic function key.

3.5.2.3 MOD 300 Operator Station

AdvaInform functions can be operated via windows in AdvaCommand. If you want to use this window feature from an Operator Station, the active Environment of the Operator Station must support these windows. How to configure support for these windows is described in the External Display Access section of the *AdvaBuild Environment Builder User’s Guide*. When the Environment supports these windows, you can open them via the Display Menu of the Operator Station.

Page links on displays built with the AdvaBuild Display Builder can also open windows into the IMS. Instructions for adding these page links to displays are provided in the *AdvaBuild Display Builder User’s Guide*.

3.6 Menus

This section describes menu items for configuring and setting up AdvaInform Basic Functions. Menu items for AdvaInform Basic Functions runtime operations are described in Chapter 4, Runtime Operation.

3.6.1 AdvaInform Menu

3.6.1.1 AdvaInform History: TTD Configuration (Master)

The TTD (Time Tagged Data) Configuration tool makes TTD log variables known to AdvaInform clients such as User API programs and SQL (SQL*Connect) clients, Figure 3-7.

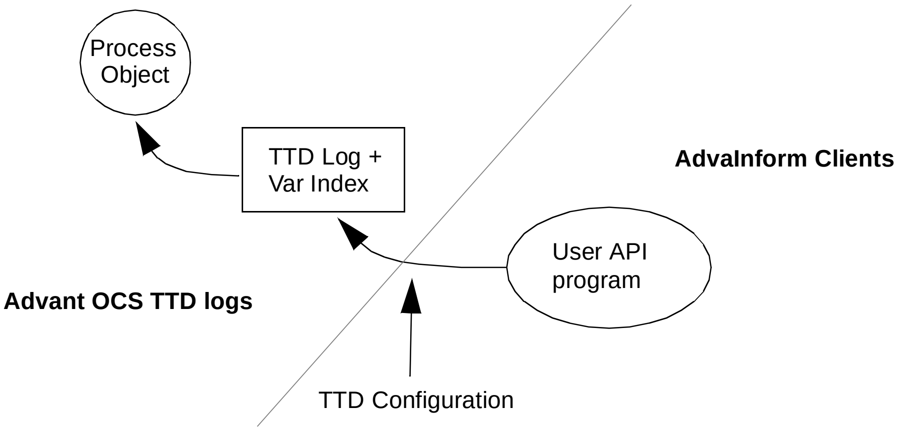


Figure 3-7. TTD Configuration

Options and features like AdvaInform Reports, Data Base Access (started from the IMS menu), Numeric Log Presentation, AdvaCommand User Interface and User API can access mapped TTD logs.

NOTE

History logging does not request TTD mapping.

To configure a TTD log to be accessible to AdvaInform clients:

1. Check the Network, Node, Log number, and Variable index of the TTD log you want to access. You can use AdvaCommand, MasterAid or MasterView to do this. Typically, the TTD log resides in Advant Controller 400 series and MasterPiece 200 series.
2. Choose **History - TTD Configuration** from the **AdvaInform** menu. This displays the TTD Configuration tool, Figure 3-8.
3. Choose a function (1 = MAP TTD, 2 = Remove Mapping, 0 = Exit)

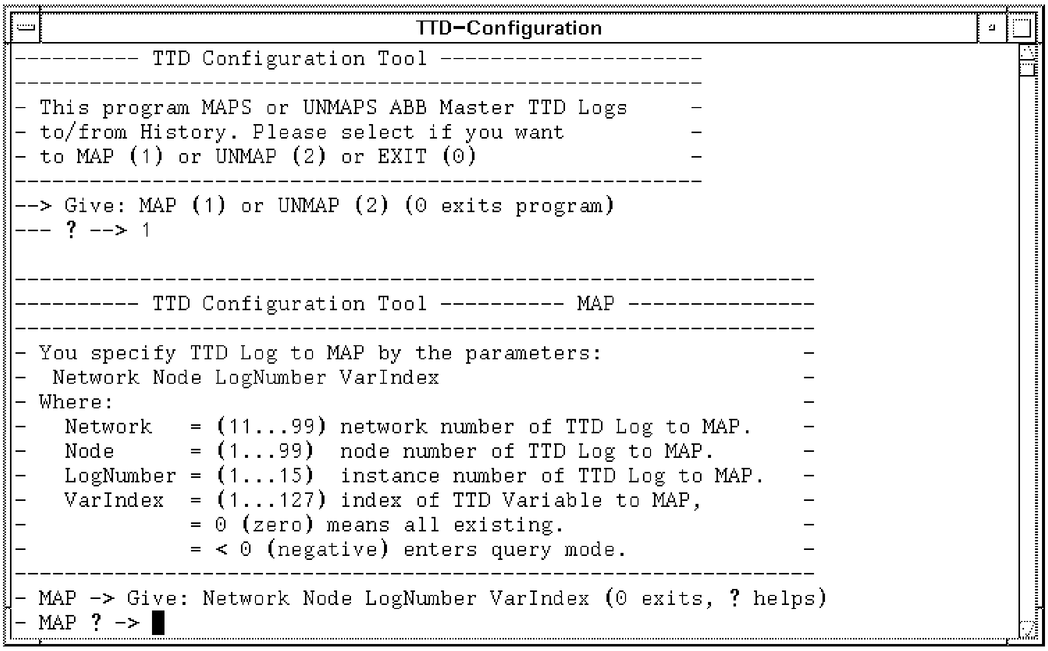
   If you choose (1) to MAP TTD (see Figure 3-8), enter the Network, Node, Log number, and Variable index you want to configure. You can configure one or several Variable instances in one session.

   Each TTD log contains several Variables. By entering -1 as parameter for Variable index, you enter query mode. The tool checks if the requested TTD log group is available for configuration.

   If the TTD log group is available for configuration and you have entered query mode, each Variable index of the TTD log is listed and you are prompted to answer y (yes) or n (no) for each variable index.

   A message is displayed to indicate if the configuration was successful or not.
   The name of the mapped TTD log to be used by SQL, User API etc is displayed.
4. Choose (2) to unmap (remove mapping). This displays the same dialog as the MAP TTD function.

   Enter Network, Node, LogNumber and Variable index of the TTD log you want to unmap. A message is displayed telling if the operation succeeded.
5. To check which TTD logs have been mapped already, you can choose **Database Access** from the **Station menu,** and access the view TTDMap. The DBLOGOBJECTNAME column contains the name that can be used as reference. The name is composed of Network, Node, Log number and Variable index.



TTD–Configuration

```
---------- TTD Configuration Tool ---------------------
-------------------------------------------------------
- This program MAPS or UNMAPS ABB Master TTD Logs     -
- to/from History. Please select if you want          -
- to MAP (1) or UNMAP (2) or EXIT (0)                 -
-------------------------------------------------------
--> Give: MAP (1) or UNMAP (2) (0 exits program)
--- ? --> 1

----------------------------------------------------------------
---------- TTD Configuration Tool ---------- MAP ---------------
----------------------------------------------------------------
- You specify TTD Log to MAP by the parameters:                -
-  Network Node LogNumber VarIndex                             -
- Where:                                                       -
-   Network   = (11...99) network number of TTD Log to MAP.    -
-   Node      = (1...99)  node number of TTD Log to MAP.       -
-   LogNumber = (1...15)  instance number of TTD Log to MAP.   -
-   VarIndex  = (1...127) index of TTD Variable to MAP,        -
-             = 0 (zero) means all existing.                   -
-             = < 0 (negative) enters query mode.              -
----------------------------------------------------------------
- MAP -> Give: Network Node LogNumber VarIndex (0 exits, ? helps)
- MAP ? -> █
```

Figure 3-8. TTD Configuration Tool

3.6.2 Settings Menu

The menu items in the **Settings** menu support AdvaInform administration and maintenance functions, See Figure 3-9.

Communication ...
User Definition ...
Station Backup ...
Process Supervision ...
Maintenance ...
System Variables ...
System Messages ..
RTA ... (1)
Date and Time ...
Node Broker... (1)

(1) Master only

Figure 3-9. Settings Menu

These functions are intended for system administrators only. To use the functions provided in this menu, you must have access at the category SYSTEM level. This section briefly describes the purpose of each function. For detailed instructions on how to use these menu items, refer to the *Advant Station 500 Series IMS User's Guide*.

3.6.2.1 Settings - Communication

Configuration of control network parameters is generally done as a part of installation as described in *Advant Station 500 Series IMS User's Guide*. You can use **Communication** to change network parameters.

3.6.2.2 Settings - User Definition

Access to and the security of AdvaInform is determined by user authority. You log on to IMS as an operating system user. This user is either a predefined user such as ocsmgr or a user defined via User Definition. For further information, refer to the introduction to user authority under Section 3.3.1, User Authorization.

3.6.2.3 Settings - Station Backup

Station Backup is handled from the Station Backup and Restore window which provides backup at different levels of your Advant Station.

3.6.2.4 Settings - Process Supervision

The AdvaInform software, including the applications you build, must be started and stopped in the right order to function properly. Therefore, when you build your own applications it is essential that you include them in the IMS start-up, shutdown and supervision.

Use **Process Supervision** to specify how to include your Applications in the IMS standard start-up, shutdown and supervision.

3.6.2.5 Settings - Maintenance

Use **Maintenance** to invoke SAM in HP-UX to add hard disk drives and printers to the IMS. To log onto SAM, you require the password for superuser (root or system administrator).

3.6.2.6 Settings - System Variables

A System Variable defines which commands to execute when you choose a menu item from the IMS Menu. The **System Variables** menu item provides the means to edit the system variables for a specific menu item. For instance, you can change the mail system or editor which is invoked when you choose the corresponding menu item in the **Station** menu. Generally, you should only edit system variables for the **Mail** and/or **Editor** menu items in the **Station** menu. It is strongly recommended that you DO NOT change system variables for any other menu items.

3.6.2.7 Settings - System Messages

This item opens the System Messages window where you can specify how many system messages can be stored by AdvaInform. Normally, there is no need to dimension the system message.

3.6.2.8 Settings - RTA (Master)

This function is used to inspect and modify the Real Time Accelerator Board. The RTA functionality is only available to IMS users of the IPA category, (ABB super user).

You must perform an Incremental Backup to save changes if you have modified RTA configurations (see *Advant Station 500 Series User's Guide*).

3.6.2.9 Settings - Date and Time

Use this menu item to change the date and time in the local node.

3.6.2.10 Settings - Node Broker (Master)

The Node Broker is a configuration tool to define the range of nodes outside your own control network with which your station shall have contact with. The Node Broker configuration is necessary to allow access (for example, subscription) towards nodes across gateway boundaries. The Node Broker must be used for this purpose whether your station is connected to a control network or to the plant network. In both cases the configuration is done in exactly the same way.

Chapter 4 Runtime Operation

4.1 Product Operation

An operator can use the installed AdvaInform Basic Functions software:

- from an X-terminal on the TCP/IP network.
- from an IMS console.
- from a PC-/MS-DOS-based computer emulating X-window.
- from an external computer using X-windows.
- from an X-window opened from an Advant Station Operator or Engineering Station.

To avoid such problems as mismatch with fonts and configuration problems it is best to use the X-monitor provided with the Advant Station, or HP's X-terminals.

Figure 4-1 illustrates the interaction between different Advant stations, and between Advant stations and other vendors' products. Using the communications and database management features included in the AdvaInform Basic Functions, operators can interact with business systems and systems for production management in the same manner as they operate process control systems. AdvaInform Basic Functions can be seen as a *server of information* and as a communication bridge between the Advant OCS and other computer systems in the plant.

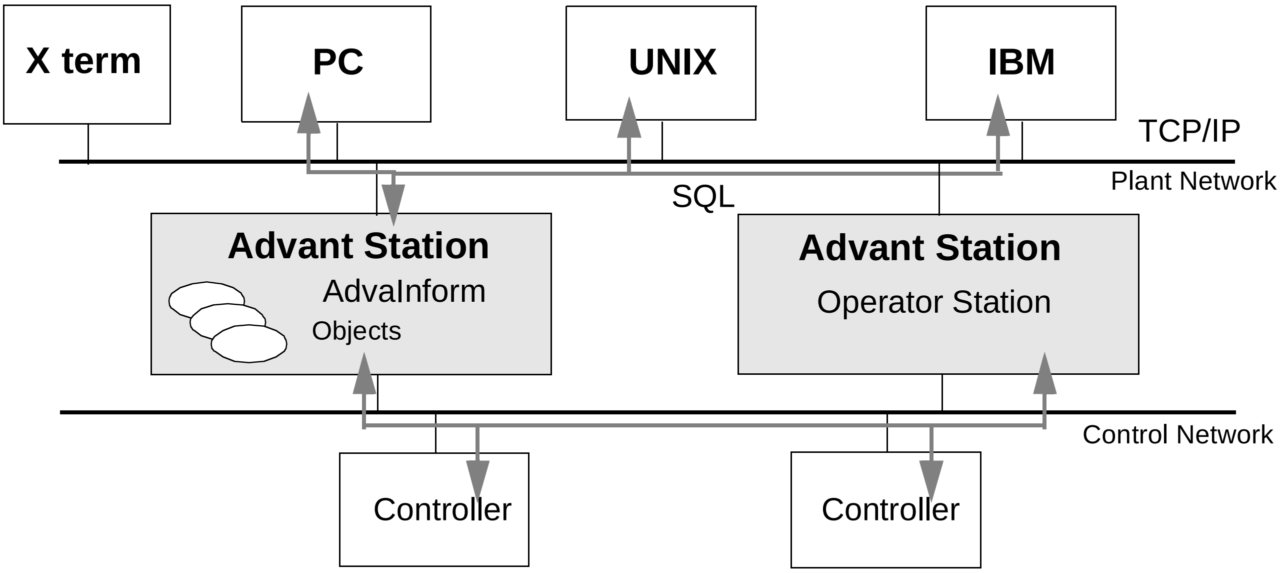


Figure 4-1. Information Exchange and Plant-wide Integration

Below, the different alternatives for information exchange with other computers/stations are briefly discussed. Details regarding configuration are described in greater detail in the *AdvaInform SQL*Connect User´s Guide*.

AdvaInform Basic Unit as Information Server to a PC

The *PC* may use a software package such as Excel to produce reports. Through SQL calls, data from one or more AdvaInform nodes or controllers can be fetched, processed, and included in such reports. Note that the required software, in this case SQL*Net and ODBC for Windows 3.11 or NT/95, must be obtained separately. See the Release Notes for the SQL*NET and ODBC products for details on installation and functionality.

All of the required database handling and communication software are included in AdvaInform Basic Functions.

AdvaInform Basic Unit as Information Server to other Database Clients

Connections to external database client computers (IBM, UNIX, VAX/VMS) are handled in the same manner as a client connection from a PC. Again, the required software for connection to the AdvaInform Station (SQL*Net for the computer/operating system concerned) has to be obtained separately, since it is not part of any licence for Advant Stations.

AdvaInform Basic Unit as Information Server to Advant Station OS

The AdvaInform Basic Unit can act as a *server of information* to an OS application, for instance the AdvaCommand Process Displays, by providing data from nodes in the TCP/IP network. A prerequisite for this kind of information interchange is that the required software SQL*Net is installed in the source nodes for data. A request from the OS for data in any database server is first received by the AdvaInform SQL*Connect server and then routed to the database server computer.

The *operator station* may have user-built process displays with information from other stations. Many types of information objects can be accessed. The most common are the basic object types - AI, AO, and so on. In controllers these are called Process Objects; and in AdvaInform they are called Basic Objects. AdvaInform User Objects such as special, customer-designed, objects built for specific purposes are also indicated in Figure 4-1. In systems with Master software, Operator Stations provide features for creating Display Elements in order to present object data, for example User Objects, on user displays.

4.2 Operating Overview

If you are not already logged in to your AdvaInform do so. See Section 3.3.3, Log In Procedure. This section also describes how to start the IMS Menu if it is not already on your screen.

Section 4.1, Product Operation lists the types of terminals and stations that can be used to access your AdvaInform software.

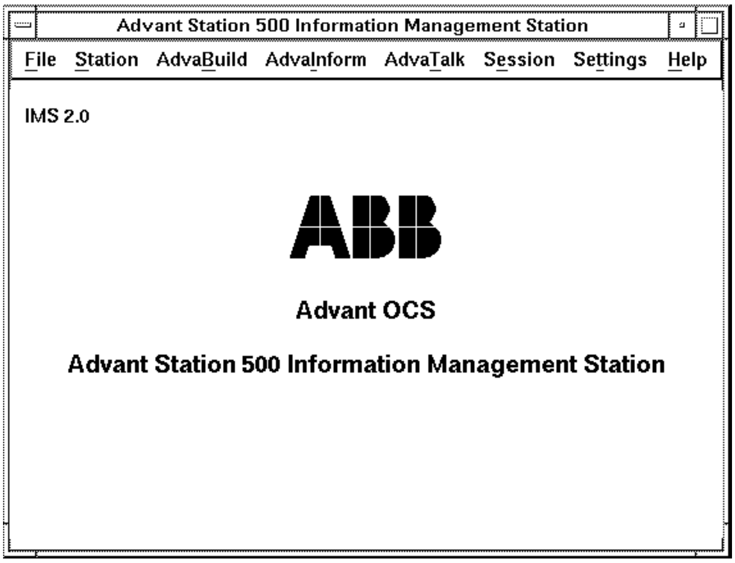

Advant Station 500 Information Management Station
File
Station
AdvaBuild
AdvaInform
AdvaTalk
Session
Settings
Help
IMS 2.0
Advant OCS
Advant Station 500 Information Management Station

Figure 4-2. AdvaInform menu

A number of functions are available via the AdvaInform menu; however, what you are allowed to do is determined by the user category you belong to. For example, if you are registered as an operator, you may use most of the functions installed. However, you may not edit the configuration of your current station. Refer to Section 3.3.1, User Authorization for a detailed description of the different user categories and their authorization.

The Menu Bar contains the following menus:

- **File**

  Use **File** - **Exit** to exit from the AdvaInform Menu.

- **Station**

  Use items in this menu to access functions included in AdvaInform Basic Function (System Status, Object Presentation, Terminal Window, Database Access, Editor, Mail, System Messages and Object Reference Maintenance).

- **AdvaBuild**

  This menu gives you access to the currently installed AdvaBuild options.

- **AdvaInform**

  The items in this menu let you access the currently installed AdvaInform functions.

- **AdvaTalk**

  The items in this menu let you access the currently installed AdvaTalk options.

- **Session**

  This menu lets you inspect your own user registration (name and category) and also to change the password you use when logging in.

- **Settings**

  From this menu the system manager normally selects the system administration tools required for configuration of peripherals, communication, creating IMS users, and so on.

- **Help**

  Use this menu to obtain help on how to perform tasks and issue commands. In addition, using Help OnVersion you can find the version numbers of the AdvaInform Basic Functions and of other software included.

4.3 Runtime Tutorial

Tutorials for applications that are accessible via the IMS menu are provided in the respective user's guides.

4.4 Operating Instructions

Accessing and using the menus is straight-forward. The mode of interaction complies with the OSF/Motif standards. All aspects of operating the AdvaInform, AdvaBuild and AdvaTalk options are described in the corresponding user guides.

The procedures described in this section require that you log in to the IMS Menu. Refer to Section 3.3, Application Start-up and User Authorization for details.

4.4.1 How to Manage the CD-ROM reader

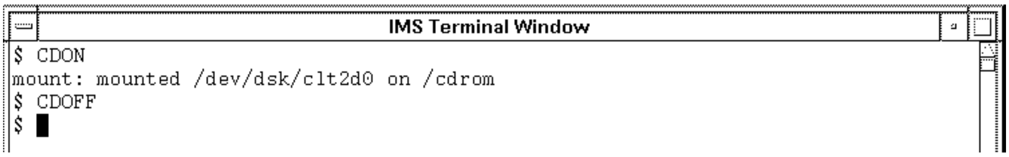
Two pre-installed scripts are provided with the AdvaInform Basic Functions for mounting and dismounting the CD-ROM reader. The scripts are designed for use in the HP-UX 10.20 environment. The scripts are:

- CDON - mounts the CD-ROM reader and makes the contents available in the directory /cdrom. To execute the script, enter **CDON** at the prompt in any terminal window as shown in Figure 4-3.
- CDOFF - unmounts the CD-ROM reader. To execute the script, enter **CDOFF** at the prompt in any terminal window as shown in Figure 4-3. Do not execute this command in /cdrom.

NOTES

Remember to insert a disk before executing the CDON script. If no disk is inserted, the CD-ROM reader is considered to be busy, and will not be accessible.

Remember to dismount the CD-ROM reader before changing disks. The disk can not be ejected while the CD-ROM reader is mounted.



```
$ CDON
mount: mounted /dev/dsk/clt2d0 on /cdrom
$ CDOFF
$ █
```

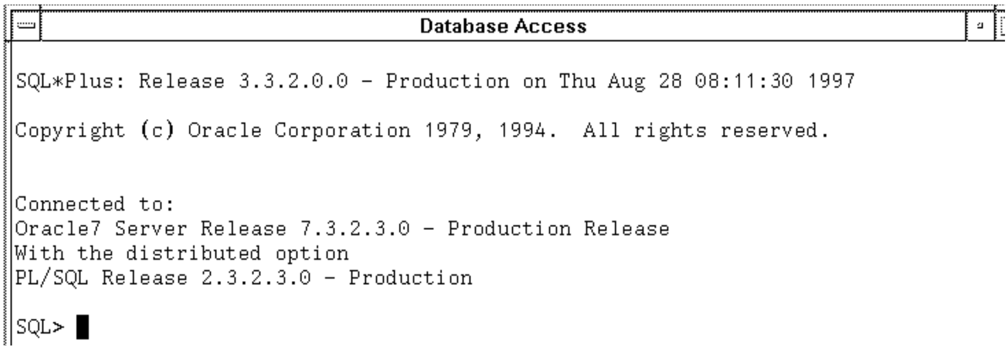
Figure 4-3. CDON/CDOFF example

4.4.2 How to Execute Queries in the Database Access Window

4.4.2.1 General

To execute SQL queries in the Database Access window:

1. Choose **Database Access** from the **Station** menu. This opens the Database Access window, Figure 4-4.



```
SQL*Plus: Release 3.3.2.0.0 - Production on Thu Aug 28 08:11:30 1997

Copyright (c) Oracle Corporation 1979, 1994.  All rights reserved.


Connected to:
Oracle7 Server Release 7.3.2.3.0 - Production Release
With the distributed option
PL/SQL Release 2.3.2.3.0 - Production

SQL> █
```

Figure 4-4. AdvaInform Database Access Window

2. To read data using SQL*Connect you should always invoke the following command before doing the first select statements.

   `set transaction read only;`

   NOTE

   The `set transaction read only` statement causes Oracle not to use any rollback segments during transactions. This prevents contention of rollback segments if one or several applications executes transactions during a long time. (The rollback segment is locked when transaction is started until a commit statement is issued).

3. Enter the query at the prompt (examples are provided in the following sections). The system will return the data.

4. To close the window when you are finished enter**: `exit`** at the prompt.

4.4.2.2 How to List All Tables, Views and Object Types for a User

To list all tables, views, and object types available to your oracle user:

1. Choose **Database Access** from the **Station** menu.
2. Enter the SQL query:

```
SQL>SELECT table_name FROM IMS_TABLES;
```

NOTE

The IMS_TABLES table describes which tables/views you are allowed to use. You can find other tables in the Advant OCS relational database, but they are only meant for internal use. Backward compatibility is not guaranteed. Do not use tables other than the tables/views described in IMS_TABLES.

4.4.2.3 How to List all Attributes and Columns for a Table or View

To list all attributes/columns available for reading in a table/view or object type:

1. Choose **Database Access** from the **Station** menu.
2. Assuming that the table AI or data_fcm is the object of interest, enter the respective SQL commands:

```
SQL>DESCRIBE AI;
```

or

```
SQL>DESCRIBE data_fcm;
```

4.4.2.4 How to Query for a Numerical Log Object

When you write a query for a numeric log object, you can use either the complete log name or the access name (log name without $HS prefix and -*n*-o suffix). The log name identifies a specific log in a composite log. The access name retrieves data from one of the logs in the Composite Log using the seamless retrieval algorithm which is described in the *AdvaInform History User's Guide*.

The following example shows how to read a number of values from a numeric log using the log name to specify a log in the log hierarchy.

```
SELECT TO_CHAR (TIME, ‘DD/MM/YYYY HH:MI:SS’), ENTRYVALUE FROM
NUMLOGVAL

    WHERE NAME = '$HSTC100,MEASURE-1-o' AND

    TIME > TO_DATE( '11 8:30:00', 'DD HH24:MI:SS')

AND

    TIME < TO_DATE( '11 9:00:00', 'DD HH24:MI:SS');
```

This query returns values for log TC100,MEASURE between 8:30 am and 9:00 am on the 11th day of the current month. More details about the TO_DATE function are provided in Appendix B, SQL Statement Reference.

The log name syntax is as follows:

`$HS`	means Historical Log.
`TC100,MEASURE`	is the log’s access name.
`1`	indicates the log number in the log hierarchy. (2 would retrieve data for the second log in the hierarchy and so on).
`o`	indicates this is original (not restored) data. If the data for this log were restored from an archive DAT, this character would be set equal to r.
`DD HH24:MI:SS`	describes the time format (DD for day, HH24 for 24 hours time, MI for minutes and SS for sec’s).

The ‘where’ clause in the example specifies both the beginning and the end of the time interval which in this case is 8:30 am to 9:00 am of the 11th day of the current month. If you do not specify a time interval, all values from the log will be returned.

The following example shows how to read the same range of values from the same numeric log as above, using the access name.

```
SELECT TO_CHAR (TIME, ‘DD/MM/YYYY HH:MI:SS’), ENTRYVALUE FROM
NUMLOGVAL
    WHERE NAME = 'TC100,MEASURE' AND
    TIME > TO_DATE( '11 8:30:00', 'DD HH24:MI:SS')
AND
    TIME < TO_DATE('11 9:00:00', 'DD HH24:MI:SS');
```

If you want to retrieve logs that are created with a specific calculation algorithm you can specify calc_alg together with the access name in the select statement. The calc_alg column is only relevant when you use the access name.

```
SELECT TO_CHAR (TIME, ‘DD/MM/YYYY HH:MI:SS’), ENTRYVALUE FROM
NUMLOGVAL
    WHERE NAME =’TC100,VALUE’ AND CALC_ALG = ‘MAX‘;
```

The following example shows how to read interpolated values from a numeric log with a time interval between the values of two hours and a time span of 12 hours. Note that you have to specify INTERVAL, TIME < xxx. and TIME > yyyy in the where clause.

```
SELECT TO_CHAR (TIME, ‘DD/MM/YYYY HH:MI:SS’), ENTRYVALUE FROM
NUMLOGVAL
    WHERE NAME = ‘$HSTC100,VALUE-1-o’ AND
    RET_TYPE = ‘INTERPOLATED‘ AND
    INTERP_ALG = ‘AVG‘ AND
    INTERVAL = 2*60*60 AND
    SELECTION = ‘CURRENT‘ AND
    TIME > TO_DATE(‘19960311 8:00:00‘, ‘YYYYMMDD HH24:MI:SS‘) AND
    TIME < TO_DATE(‘19960311 20:00:00‘, ‘YYYYMMDD HH24:MI:SS‘) ;
```

4.4.2.5 How to Query for a Message Log Object (MOD 300 example)

CCF, TCL, console, alarm acknowledgment, and log on messages are stored in the message log. This log has the access name MODMSGLOG_x_y where where x and y are the device and subdevice addresses of the node. This message log can also be accessed by the non-node specific name: MODMSGLOG. This aids in developing Oracle applications that are node independent. Before you can access the message log, it must be configured via AdvaInform History as described in the *AdvaInform History User's Guide*.

The principle attributes are listed in Table 4-1.

Table 4-1. MODMSGLOG Attributes

Column Name	Format	Description
IDX:	NUMBER	Internal index column.
SOURCE_NODE	NUMBER	Node address message came from.
NETWORK:	NUMBER	Subdevice
LOCALTIME	DATE	Time of the message (one second resolution).
UTC	NUMBER	utc of the message in seconds.
MICROSECONDS	NUMBER	Micro second portion of time of the msg.
MESSAGE_TYPE	NUMBER	Type of message: 0 = log on/log off messages 1 = alarm messages 2 = billboard messages 3 = parameter change messages 4 = ladder logic messages 5 = Statistical Process Control (SPC) messages 6 = Alarm acknowledgment messages 7 = Diagnostic messages 8 = Batch 300 messages
MESSAGE_COUNT	NUMBER	Not implemented.
PRIORITY	NUMBER	Not implemented.
UNIT_NAME	CHAR(12)	Unit message originated from.
TAG_NAME	CHAR(32)	Tag message is for.
BATCH_NAME	CHAR(40)	Batch message is associated with.
SEQUENCE_NAME	CHAR(12)	Sequence that generated the message.
MESSAGE_STATUS	NUMBER	Status of the message entry.
MESSAGE	CHAR(255)	The message text.

Example queries for MODMSGLOG are provided below.

Example 1 - General Query

This query retrieves the tag and message for all messages in the message log:

```
SELECT TAG_NAME, MESSAGE
FROM MODMSGLOG;
```

Example 2 - Alarm Message Query

This query retrieves tag and message for all alarm messages:

```
SELECT TAG_NAME, MESSAGE
FROM MODMSGLOG
WHERE MESSAGE_TYPE=1;
```

Example 3 - Parameter Change Query

This query retrieves tag and message for all parameter change messages:

```
SELECT TAG_NAME, MESSAGE
FROM MODMSGLOG
WHERE MESSAGE_TYPE=3;
```

Example 4 - Query for Messages Prior to Specified Date and Time

This query retrieves local time and message for all messages prior to January 1 1995:

```
SELECT to_char (LOCALTIME, ‘mm/dd/yy HH24:MI:SS’),
MESSAGE
FROM MODMSGLOG
WHERE LOCALTIME < to_date (‘1995-01-01’, ‘YYYY-MM-DD’);
```

Messages from AdvaInform Basic Objects can be read from the message log with access name EAMSGLOG in table BCHS_MESSAGES. Use BCHS_MESSAGES in the FROM clause, and use the full log name $HSEAMSGLOG-1-o for LOG_NAME. For this log, an event reason is returned in addition to the log name, name of the object that caused the message, and local time.

4.4.2.6 How to Present Objects and Logs

The following object presentation displays are available:

Common object displays: AI, AO, DI, DO, DAT, TEXT

Master specific object displays: PIDCON, RATIOSTN, MANSTN

Common log object displays: NUMERIC LOG and MESSAGE LOG

Master specific log object displays: Local EVENT LOG, EVENT LOG

You can open the object presentation for a specific instance of any one of these object types to display data and status information for the object instance.

When the local Basic Objects included in AdvaInform Object Handling generate events, those can be stored locally. This requires AdvaInform History to be installed and that the message log EAMSGLOG is configured via AdvaInform History as described in the *AdvaInform History User's Guide*.

NOTE

You will not be able to pick up an instance belonging to AdvaInform Object Handling until it is started.

To open an object or a log presentation:

1. Choose **Object Presentation** from the **Station** menu.

   This displays a submenu with the object and log presentation options Figure 4-5.

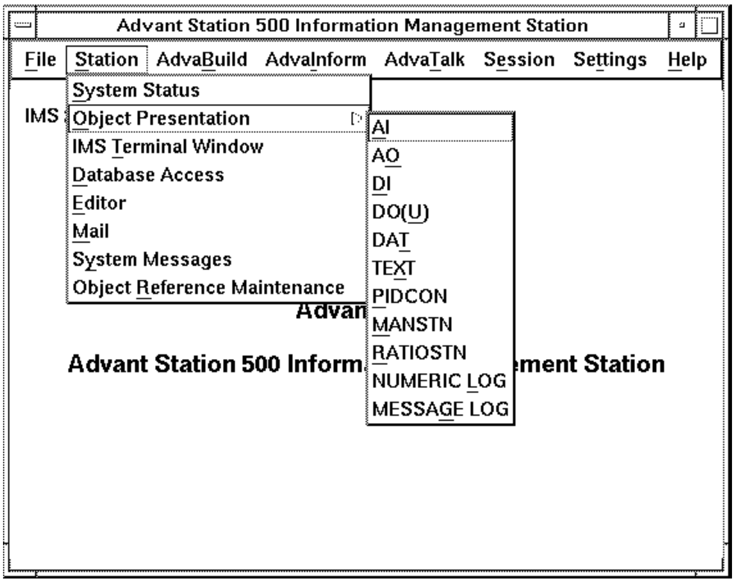


Figure 4-5. Submenu for Object Presentation (Master)

2. Choose the object or log type for the object or log instance that you want to open.

   This displays an empty object or log presentation, Figure 4-6.

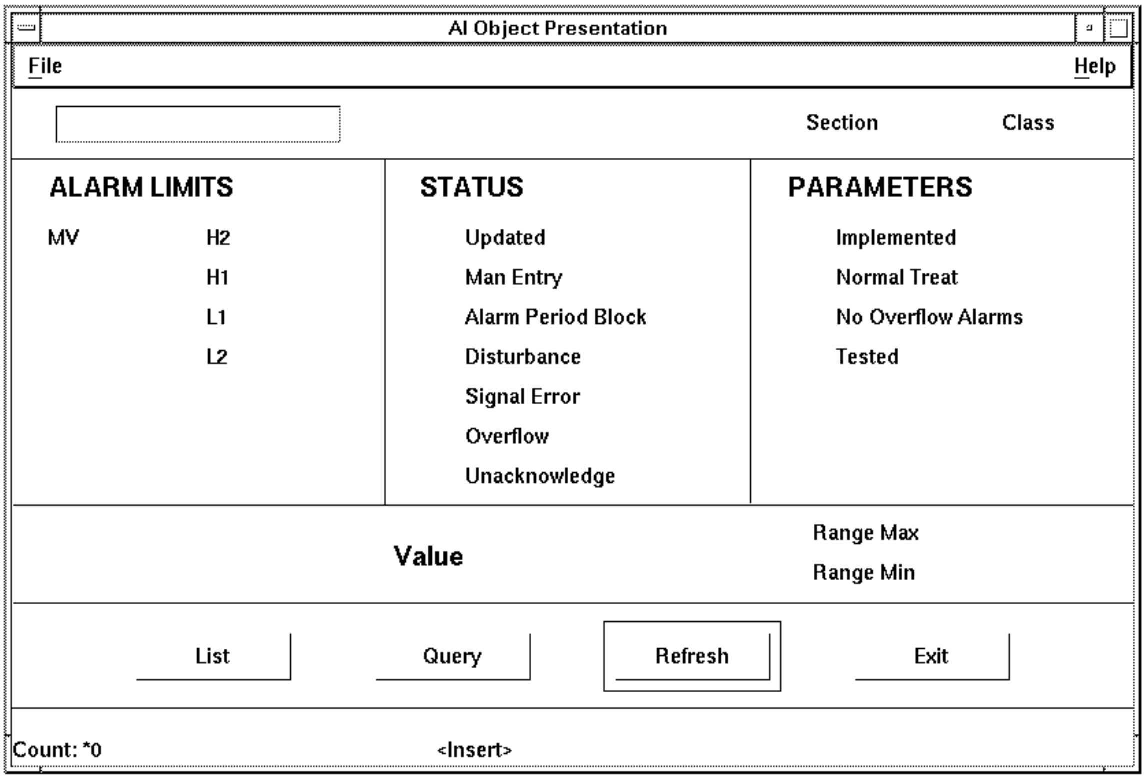
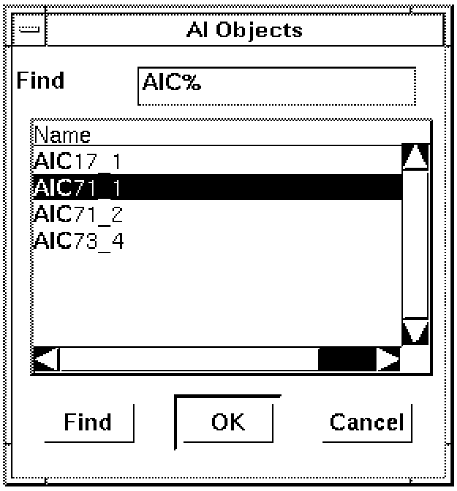


Figure 4-6. Empty AI Object Presentation

3. There are two ways to query for an object or log.

   a. For objects/logs that are already resolved, the easiest method is to select via the **List** button.

      Click the **List** button to open the selection box, Figure 4-7. This selection box lists all resolved objects/logs that can be selected. Use the scroll bar to scroll through the list. Select an object/log and click **OK** or double click on the name. If there are too many objects/logs in the list you can search for objects or logs with a specified filter parameter. Write the filter parameter in the Find field and then click **Find**. You can use % as a wildcard in place of one or several characters.



Find: AIC%

AIC17_1
AIC71_1
AIC71_2
AIC73_4

Figure 4-7. Objects selection box

The selection box has the following fields and push buttons:

Find field	A field where you can specify the filter.
Find button	Click on **Find** to search for objects/logs according to the specified filter.
OK button	Click on **OK** to present the selected object/log.
Cancel button	Click on **Cancel** to close the dialog box without selecting anything.

b. If the object or the log is not in the list, click **Query** and enter the complete object/log name in the name field. Click **Query** again or press <**Return**> to start the query. This is useful when the object/log has not been resolved yet.

4. Click on **Refresh** to present the selected object/log once more with the most recent data.
5. Click on **Exit** to leave the object/log presentation.

Common Menu Item for Object and Log Presentation

The following are common menu items for Object and Log Presentations.

File -	Exit	Exit the form
Help -	Keys	Lists all available function keys
	display Error	Shows last error from Oracle.

Common Push Buttons for Object and Log Presentation

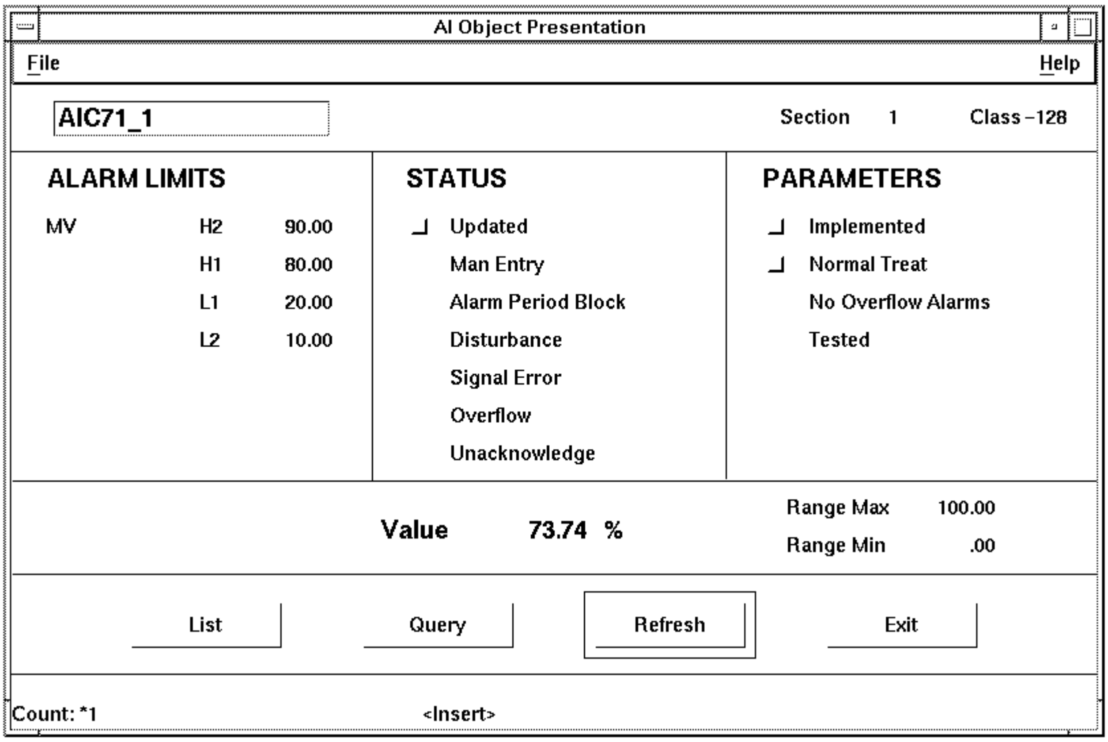
The following are common push buttons for Object and List Presentations.

List	Click on **List** to open the Objects/logs selection box, Figure 4-7, where all resolved object/logs are listed. Double click on the object/log or use **OK** to select the object/log. Use the scroll bar to scroll through the list.
Query	Click on **Query** to clear all fields and put the marker in the object/log name field. Fill in the object/log name and click **Query** again, or press <**Return**>, to start the query.
Refresh	For Objects, click on **Refresh** to present the selected object/log once more with the most recent data.
Reset	For Logs click on **Reset** to clear all fields and set Local Time to the default time span, from present time and one hour backward
Exit	Click on **Exit** to leave the object/log form.

Items Common to Log Presentation Only

The following is a summary of items that are common to all log presentations.

Sort by	Specifies which field the presentation will be sorted by. You can achieve this by clicking on any of the "Sort by" buttons available in the presentation. If Local time is selected the presentation will be sorted by time with the oldest item first. If a text field, for example Object Name, is selected the presentation will be sorted in alphabetic order. If a numeric field is selected the presentation will be sorted in numeric order.
Criteria	Specifies the selection criteria. For example, Local Time specifies the time span that will be presented. All fields can be entered to reduce the presentation list. Empty fields means all items. Click on the button next to the field you are interested in. This will bring up a dialog box where you choose a reason by selecting it and then click on **OK** button. This will take you back to the presentation. Click on **Query** button at the bottom of the presentation.
Scroll bar	Presentations display scroll-bars size is set according to the amount of records that are present in memory after display start-up. This corresponds to two pages and does not correspond to the total amount of records.



AIC71_1

Section 1 Class -128

ALARM LIMITS

MV	H2	90.00
	H1	80.00
	L1	20.00
	L2	10.00

STATUS

- ✓ Updated
- Man Entry
- Alarm Period Block
- Disturbance
- Signal Error
- Overflow
- Unacknowledge

PARAMETERS

- ✓ Implemented
- ✓ Normal Treat
- No Overflow Alarms
- Tested

Value 73.74 %

Range Max 100.00
Range Min .00

Count: *1 <Insert>

Figure 4-8. Example of a Log Presentation

See Appendix C, Object Presentations for details on each Forms Display.

4.4.3 How to Use System Status

General Description

System Status is started from the **Station** menu. The System Status function has three basic displays:

- Overview (Master only)
- Network Status (Master only)
- Node Status

The basic layout of the System Status display is shown in Figure 4-9.

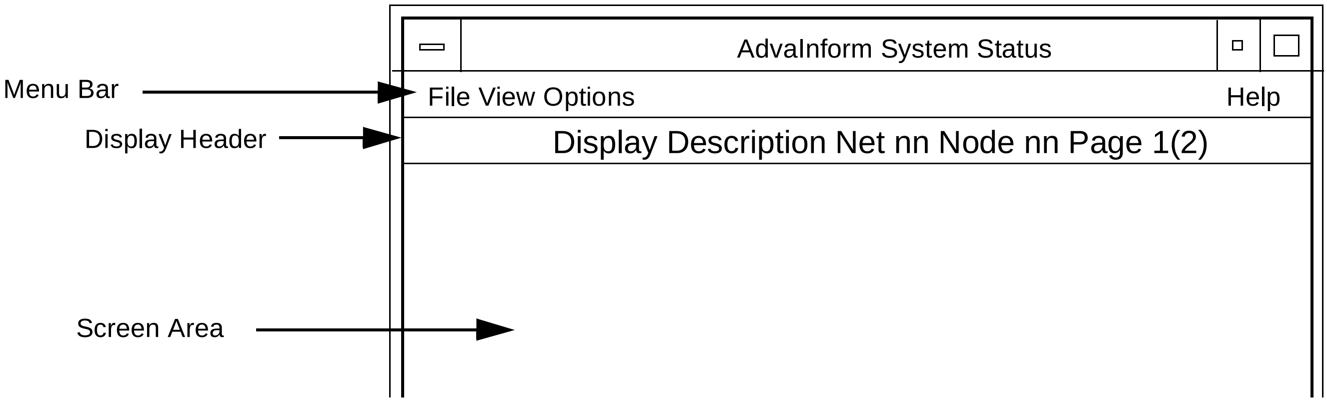


Figure 4-9. System Status Display Layout

The **View** and **Option** menus let you select what information to show. The **Help** menu displays help information about the System Status function.

The Display header shows the name of the active display, description, network and node number of node from which the data is fetched. When applicable (for overview and network displays), the page number is shown.

The information shown in the screen area depends on the active display. If the display area is larger than the window size, scrollbars are provided to view hidden parts of the display. The information in the displays is organized in rows. In the Overview and Network displays each row contains information about a node.

Row selection is done via the mouse. Selection is indicated by a rectangle that is drawn round the selected row, Figure 4-10. A second mouse click on a selected row deselects the row. When a node is selected the menu items in the **Action** menu become available and can be used to select various status display.

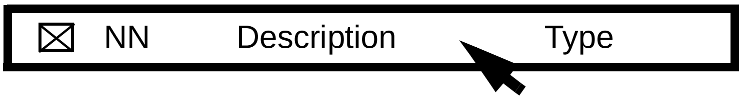


Figure 4-10. Selection rectangle

System Status Displays - Common Functionality:

Some functions, like the menus, are common to all of the status displays. The menu items are briefly described below:

File -	Exit	Close the system status window.
View -	Overview	Shows a system overview of the nodes connected to the same network as the IMS node. The display can show information for 50 nodes. If the network contains more nodes, they can be shown via the **Second Page** menu item.

	Node Status	Displays the state of a node in the same network as the IMS.
	Network Status	Shows network status of the nodes in the network where the IMS resides. A maximum of 90 nodes can be presented at the same time. If the network contains more nodes they can be displayed via the **Second Page** menu item.
	First Page	The overview and network status displays are divided into two parts to make it possible to display networks with a large number of nodes. This menu item selects the first part of the display.
	Second page	The overview and network status displays are divided into two parts. This menu item selects the second part.
Options -	Node description configuration	Lets you define a node description for external nodes (not part of the Advant OCS). This menu item is only available in the System Overview display.
Help -	On Help	Provides information on how to use the Help menu.
	On Window	Describes how to operate the current window.
	On Keys	Provides information about the keyboard, for example, how to use the Accelerator keys.
	On Index	Provides information about System Status specific topics - keywords - by double-clicking on the desired entries.
	On Version	Provides information about version numbers for certain software in your IMS, for example, the AdvaInform Basic Functions.

To configure the description (name) of Master nodes, choose **Node Description Configuration** from the **Options** menu. This displays the System Overview Configuration display, Figure 4-11. For Advant Station nodes, this information is stored in the nodes themselves and is automatically obtained by the System Overview display. Configuration can only be done from the System Overview display

The System Overview configuration dialog has one text entry field for node number and one text field for the description. Use the **Tab** key to move between fields. The node number can be a value between 1 and 99 or the value 127 (gate).

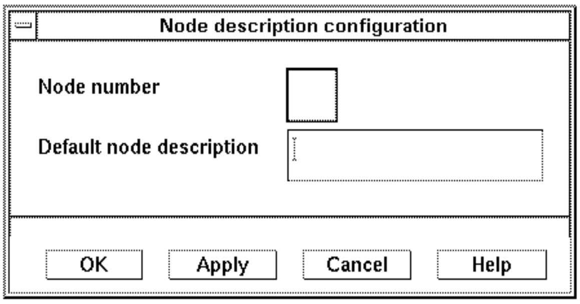


Figure 4-11. Node description configuration display

Node Status Display

The node status display shows status for a number of vital items in a node. The information varies depending on the node type. The Systems Status function displays specific node status information for IMS, OS, AC and Master nodes. The Node Status display for IMS is available both in Master and in MOD300.

The information is presented in separate scrollable columns as shown in Figure 4-12 and as described below. The rows in these columns are not selectable.

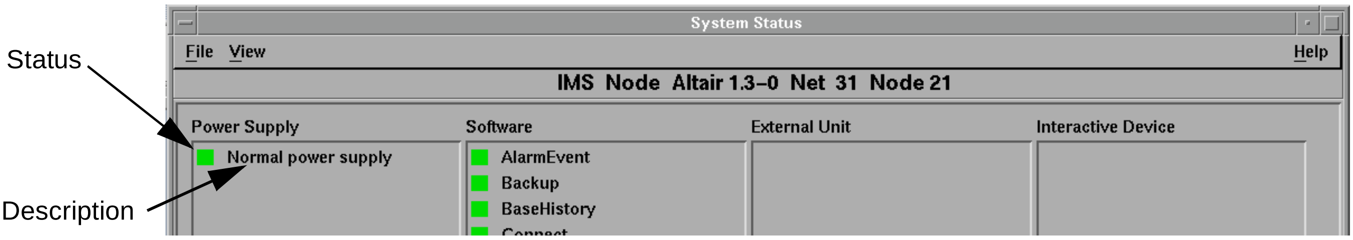
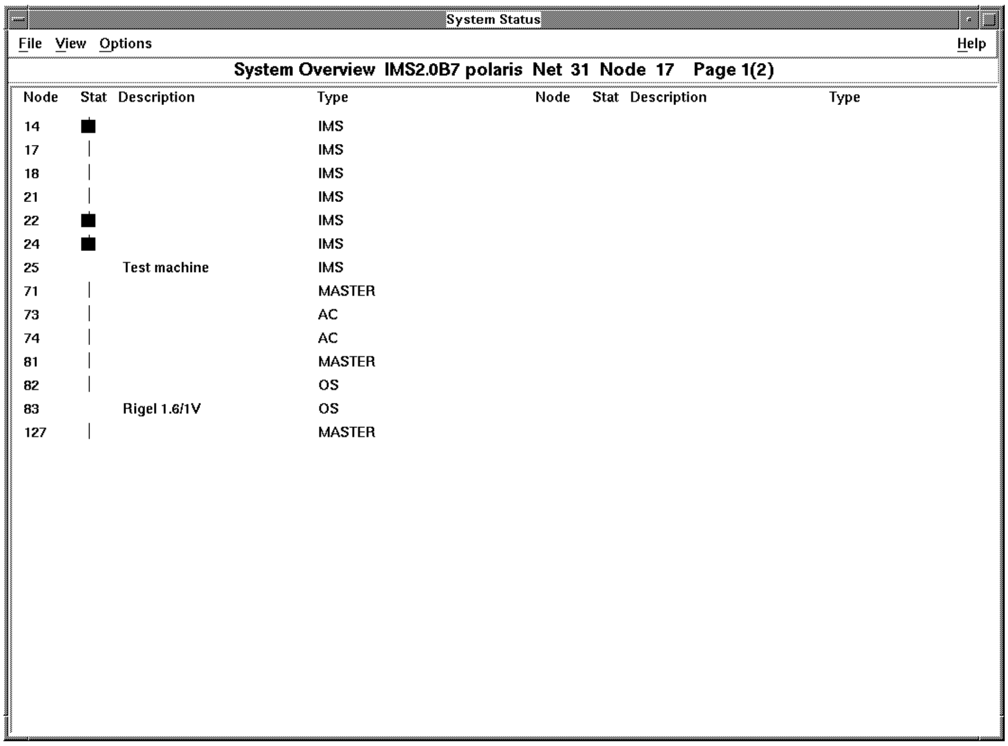


Figure 4-12. Example, Node Status Display

Status	Filled green box indicates correct operation Outlined green rectangle indicates a stand-by state Outlined yellow cross indicates a warning Outlined red cross indicates a fault
Description	Description of the displayed status item

4.4.3.1 System Overview Display

The System Overview display is shown in Figure 4-13



System Overview IMS2.0B7 polaris Net 31 Node 17 Page 1(2)

Node	Stat	Description	Type	Node	Stat	Description	Type
14	■		IMS				
17	|		IMS				
18	|		IMS				
21	|		IMS				
22	■		IMS				
24	■		IMS				
25		Test machine	IMS				
71	|		MASTER				
73	|		AC				
74	|		AC				
81	|		MASTER				
82	|		OS				
83		Rigel 1.6/1V	OS				
127	|		MASTER				

Figure 4-13. System Overview display

The System Overview display provides an overview of the nodes connected to the same network as the IMS node. Up to 50 nodes can be displayed in the first page of the display. If there is more than 50 nodes in the network, choose **Second Page** from the **View** menu to show 50 additional nodes.

The following information is provided for each node in the display:

Node (Number)	This column indicates the node number for nodes in the same network as the IMS. Network number is not displayed.
Stat (Summary status)	This column provides a summary of all statuses in a node. If any status has a fault, the summary status for the node will indicate that. The Node status display can then be used to determine the cause of the problem. The sum of all network statuses in the node is also included in the summary status. This means that if a node goes down in the network, all nodes that have communication with that node get a warning summary status. The summary status is indicated with the following icons: • A filled green box indicates normal status. • An outlined yellow box with a cross indicates a warning state in the node. • If no box is displayed the node has active communications with the network but doesn’t answer requests for summary status. An example of this state is when a Controller node is in configuration state
Description	For Advant Stations and Advant Controller 400 series, descriptions are fetched from the corresponding configuration files. For other nodes, you can define your own node descriptions. See the description below for details. The color of the description text is the same as the summary status, or black if the summary status is unavailable.
(Node) Type	OS, IMS or ES, indicate Advant Station nodes and AC indicate Advant Controllers. Non-Advant Station nodes are referred to as MASTER. If the node type can not be obtained for an Advant node, the type is displayed as ADVANT. It is not possible to display the node status for a node whose node type is ADVANT.

4.4.3.2 Node Status Display IMS

The Node status display, Figure 4-14, shows status for Advant hardware and software, for example, the state of software processes, mass memories and printers.

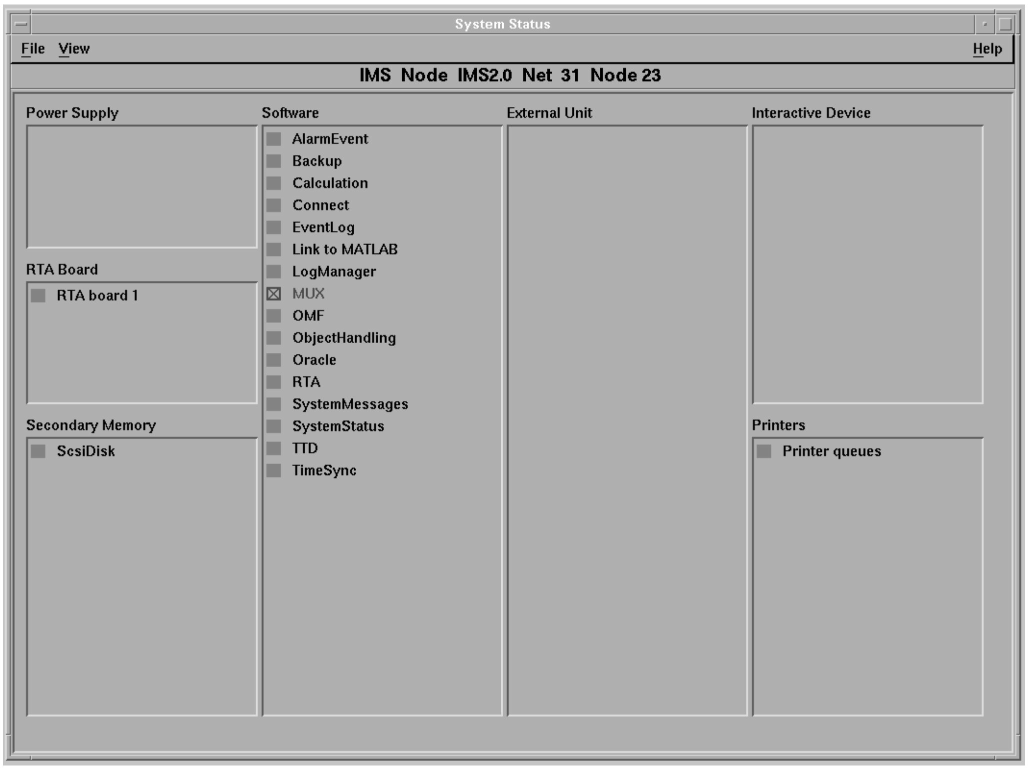


Figure 4-14. Node Status Display

The most important status information displayed by the Node status display are:

Mass Memory	Shows the status of devices (hardware), for example, mass memories and screens. Hardware supervision is done automatically by the AdvaInform Basic Functions. The user can not define devices to be supervised.
Software	Shows the status of software processes configured for supervision. Processes included in the AdvaInform Basic Functions plus the AdvaInform, AdvaBuild and AdvaTalk options are automatically defined for supervision when they are installed. You can define User processes for supervision by the Node status function.
Printers	Status for printers and printer queues.

4.4.3.3 Node Status Display Advant Controller (Master)

The Node Status display, Figure 4-15, indicates the state of a Controller node. The information is the same as when you click on the **ADVANT Controller 400** dynamic key on an OS. The OS however has additional node status displays for AC nodes that are not available in the IMS.

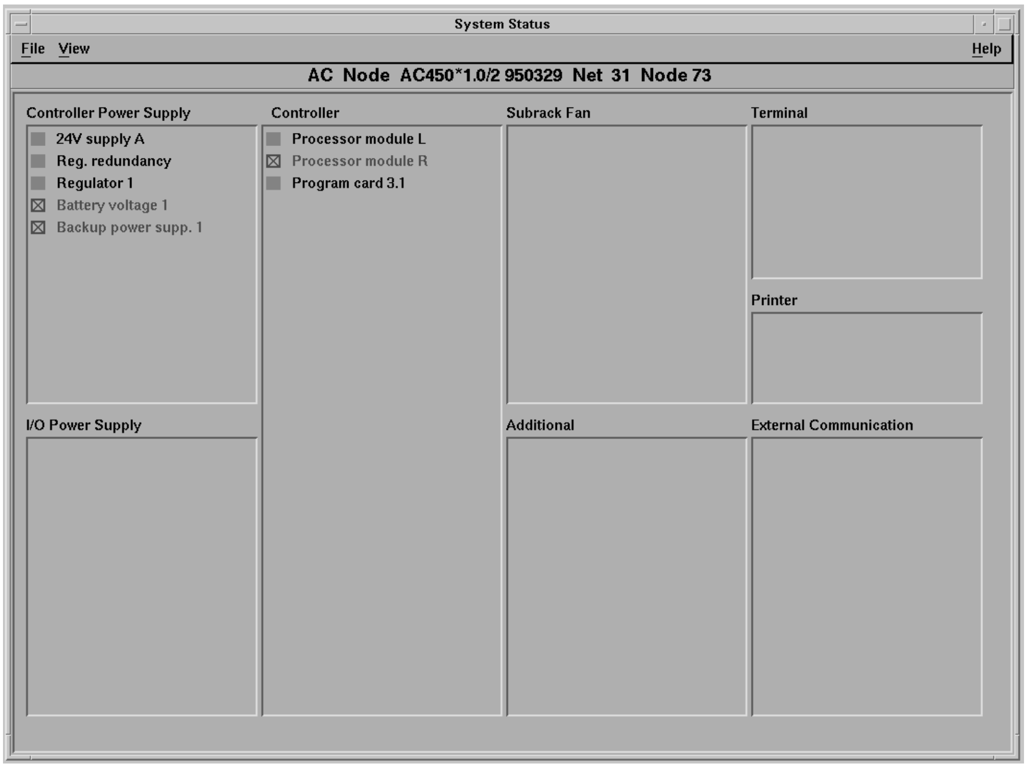


Figure 4-15. Node Status Display for Advant Controller

4.4.3.4 Node Status Display - Master Nodes

The Master Node Status display, Figure 4-16, indicates the state of a Master node. It can be any type of Master node, for example: MasterPiece, MasterView or SuperView. The OS has additional node status displays for MasterPiece nodes that are not available in the IMS.

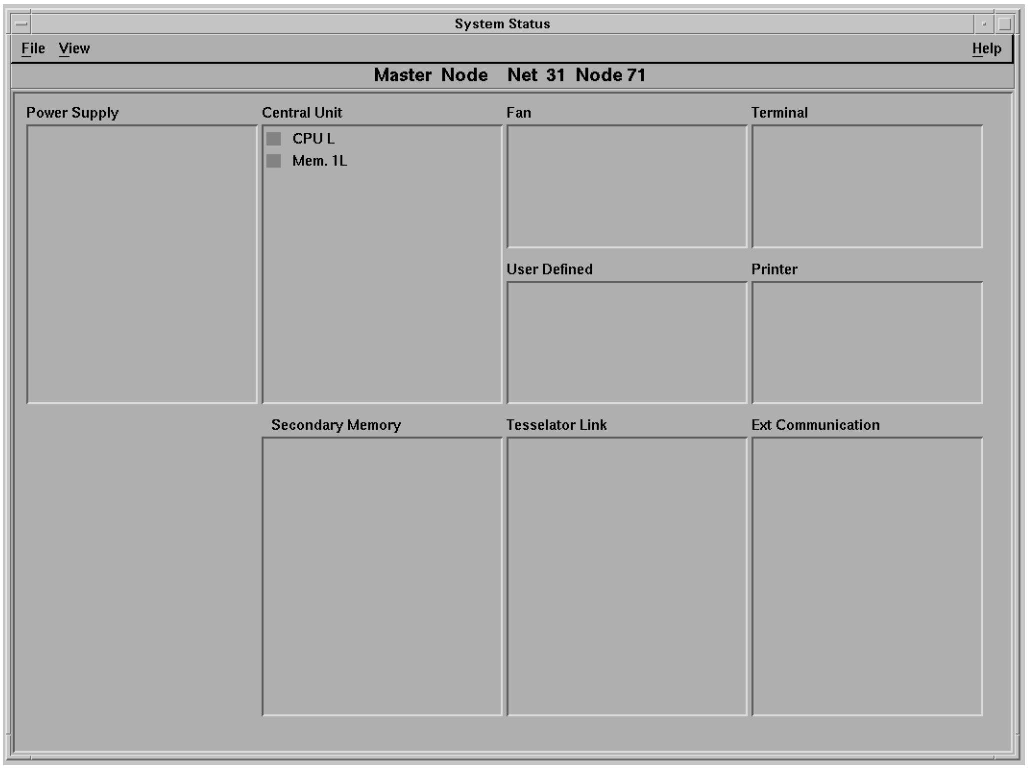


Figure 4-16. Node Status Display for Master Nodes

4.4.3.5 Node Status Display Operator Station (Master)

The OS Node Status display, Figure 4-17, shows information about a OS node. The displayed information is the same as can be obtained from the OS System Status function.

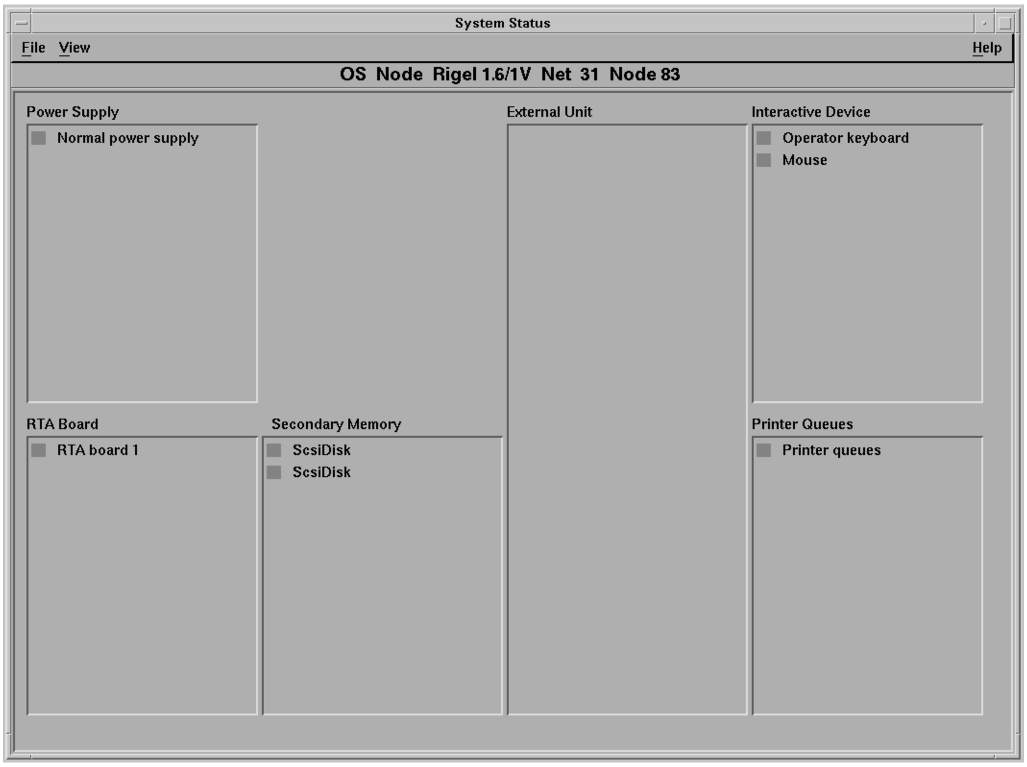
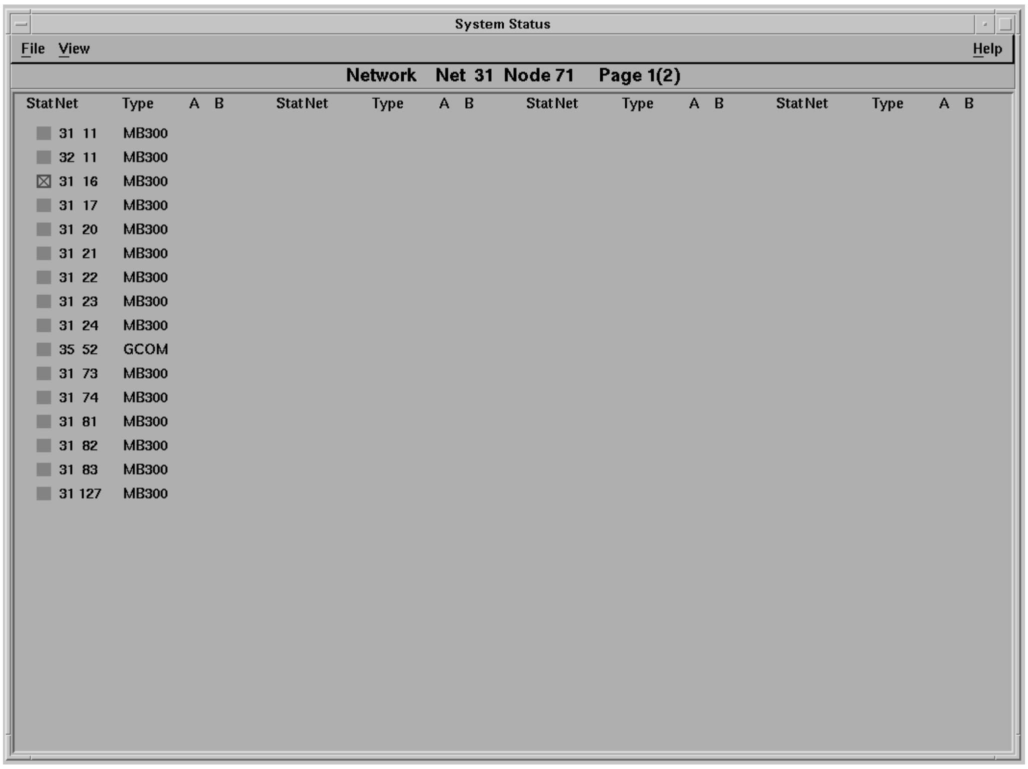


Figure 4-17. Node Status Display for OS

4.4.3.6 Network Display (Master)

The Network status display, Figure 4-18, shows whether or not a node can be accessed through the Control or Plant Network



Network Net 31 Node 71 Page 1(2)

Stat	Net	Type	A	B
■	31 11	MB300		
■	32 11	MB300		
☒	31 16	MB300		
■	31 17	MB300		
■	31 20	MB300		
■	31 21	MB300		
■	31 22	MB300		
■	31 23	MB300		
■	31 24	MB300		
■	35 52	GCOM		
■	31 73	MB300		
■	31 74	MB300		
■	31 81	MB300		
■	31 82	MB300		
■	31 83	MB300		
■	31 127	MB300		

Figure 4-18. Network Display

Nodes are represented with one or two rows of data, depending on if the node has redundant communication cards. Each row contains the following information:

Stat	This column contains status boxes that show the network status of one communication device for the node. Redundant nodes are shown twice with different network numbers. • A filled green box indicates that network communication is OK. Advant Station nodes are always indicated as green unless they have been turned off or been physically disconnected from the network • A outlined red cross indicates complete loss of communication with one of the nodes communication cards.
Net	This column indicates the network and node number.

Type	This column indicates the type of communication protocol, for example: MB300, MB200 and GCOM. UNKWN denotes that the node type, for some reason, is not known to the status display subsystem. When UNKWN is indicated, you should investigate to determine the cause.
A, B	These two columns indicate the status of redundant MB200 communication devices.

The Network display provides the means to inspect the status of other nodes as seen from another node in the same control network. This is done as follows:

1. On the list, click on the node you want to “look” from

   The selected node has to be in the same control network. If not, an error message is displayed.

2. Choose **Network** from the **View** menu.

If the network contains more than 90 nodes, you can display the second network page by choosing **Second Page** from the **View** menu.

4.4.4 How to Access the Relational Database

Instructions with specific examples for entering queries are provided in Section 4.4.2, How to Execute Queries in the Database Access Window.
For detailed information on specific SQL*Plus commands, refer to Appendix B, SQL Statement Reference.

To get on-line help for a specific command, enter **`Help`** on the command line. To view the syntax for SELECT enter:

```
SQL>HELP SELECT
```

Since SQL*Plus is used under the Oracle Run-Time Licence, help text is only provided for allowed SQL commands.

To leave or close the database access window, use the window pull-down menu in the upper, left corner and then choose the **Close** option. Another way to close the window is to enter the command **exit** (after the prompt).

4.4.5 How to Edit

The default editor for Advant Station 500 Series is HP CDE Text-Editor. Other editors may be made available from your system manager. For more information see the *Advant Station 500 Series IMS User´s Guide*.

To leave or close the text editor, use the window pull-down menu in the upper, left corner and choose the **Close** option.

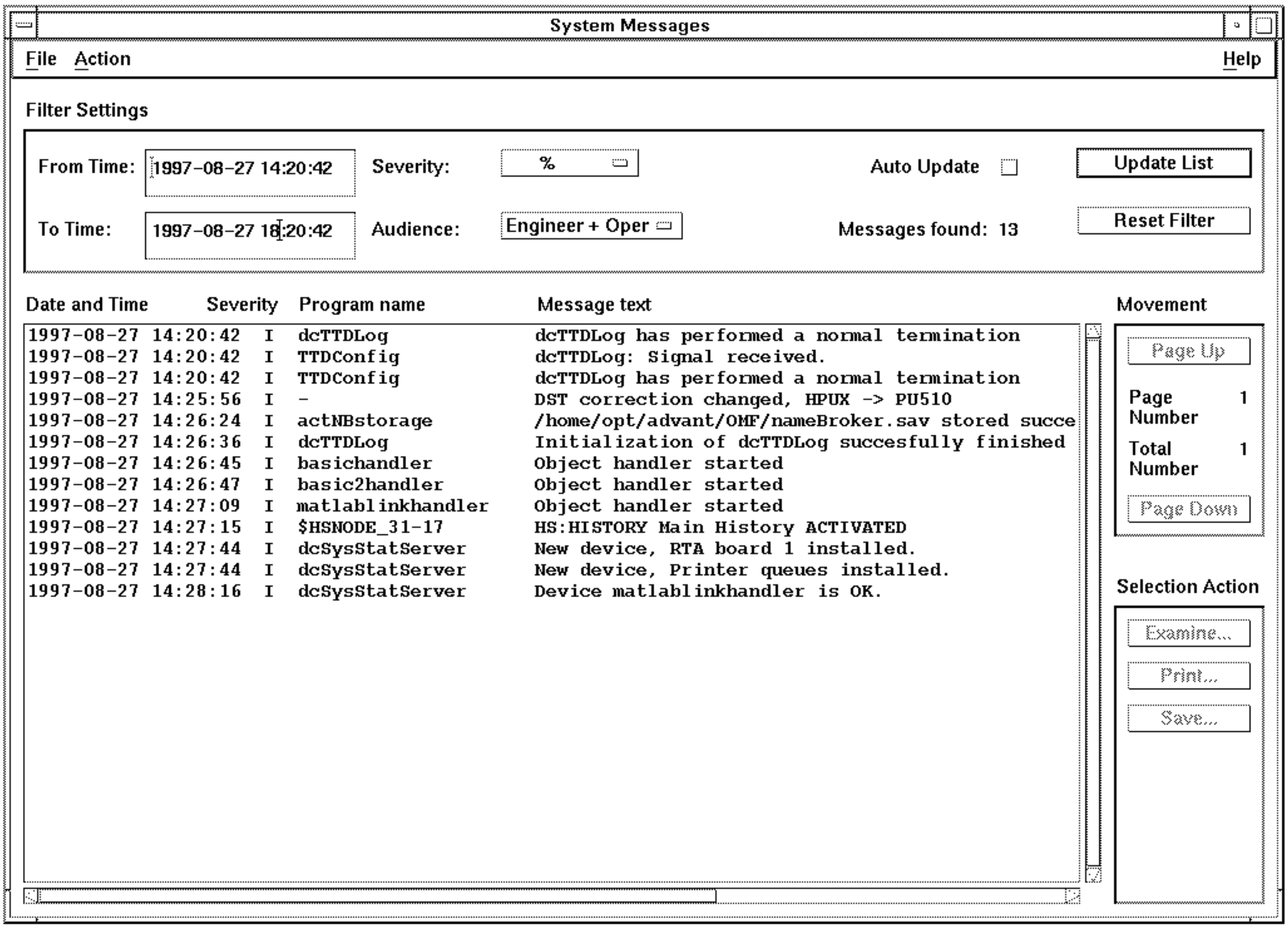
4.4.6 How to Send Mail

The default mail package for Advant Station 500 series is HP-elm.

Other mail packages may be made available from your system manager. See *Advant Station 500 IMS User´s Guide* for details on mail.

4.4.7 How to Analyze System Messages

Choose **System Message** from the **Station** menu to display the System messages window, Figure 4-19.



From Time: 1997-08-27 14:20:42
To Time: 1997-08-27 18:20:42
Severity: %
Audience: Engineer + Oper
Auto Update ☐
Messages found: 13
Page Number 1
Total Number 1

```
Date and Time        Severity  Program name        Message text
1997-08-27 14:20:42  I  dcTTDLog            dcTTDLog has performed a normal termination
1997-08-27 14:20:42  I  TTDConfig           dcTTDLog: Signal received.
1997-08-27 14:20:42  I  TTDConfig           dcTTDLog has performed a normal termination
1997-08-27 14:25:56  I  -                   DST correction changed, HPUX -> PU510
1997-08-27 14:26:24  I  actNBstorage        /home/opt/advant/OMF/nameBroker.sav stored succe
1997-08-27 14:26:36  I  dcTTDLog            Initialization of dcTTDLog succesfully finished
1997-08-27 14:26:45  I  basichandler        Object handler started
1997-08-27 14:26:47  I  basic2handler       Object handler started
1997-08-27 14:27:09  I  matlablinkhandler   Object handler started
1997-08-27 14:27:15  I  $HSNODE_31-17       HS:HISTORY Main History ACTIVATED
1997-08-27 14:27:44  I  dcSysStatServer     New device, RTA board 1 installed.
1997-08-27 14:27:44  I  dcSysStatServer     New device, Printer queues installed.
1997-08-27 14:28:16  I  dcSysStatServer     Device matlablinkhandler is OK.
```

Figure 4-19. System Message

The **Auto Update** button lets you select manual or automatic refresh of the system message list. **Page Up**, **Page Down** and **Update List** push-button provide the means to step forward and backward through the list, and to refresh manually.

The fields and buttons on this display are described below.

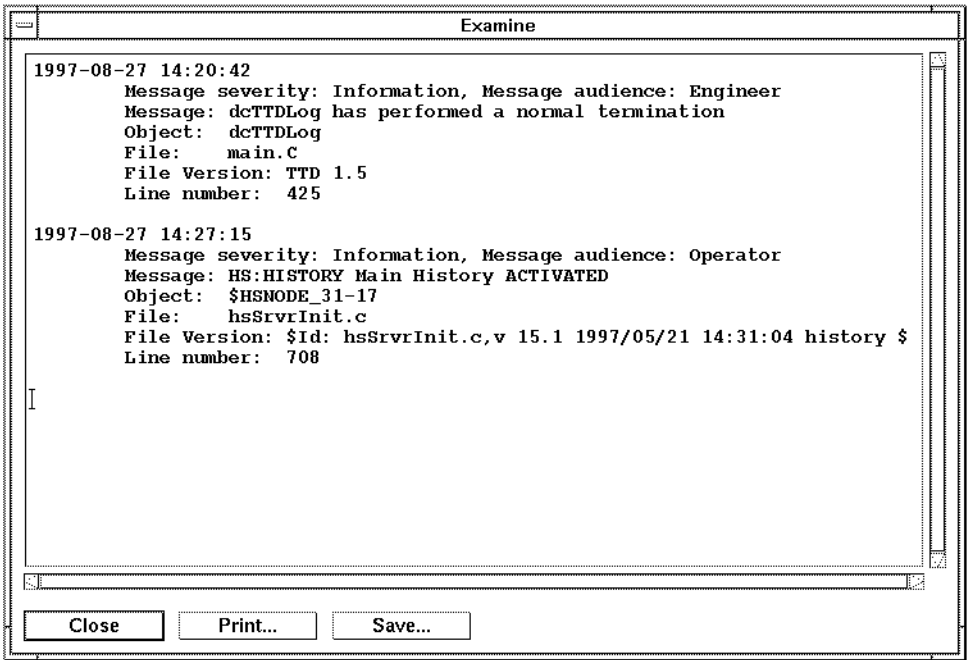
From Time / To Time	From Time and To Time fields define the time span for system messages to be displayed. Use the **Clear** button to clear the fields. Double-click on a system message to set the values in the time fields to the time of the log involved. The full format (both date and time) for From and To Time is: year-month-day hour:minute:second
Severity	Used to select system messages of a certain severity level. The character“%” selects all levels. To select the messages of a specific severity level, select either Information, Warning, Error or Fatal from the Severity option button. **Information** means Informational Errors. Normally no action is required. **Warning** means an event has occurred that *might* have to be investigated. **Error** means an error has occurred that impairs a function but does not stop the process. **Fatal** means an error has occurred that has stopped a vital process. The affected function can not be used without your system manager investigating the matter and restarting the process involved.
Audience	Specifies the audience the system messages are directed to. The following values can be used, Operator, Operator and Engineer, and “%” (all audiences).
Reset Filter	Click on Reset Filter to clear all settings in all display fields and insert “%” instead. It also resets Automatic Update.
Auto Update	Click on Auto Update to dim the Page Up and Page Down buttons and lock the From Time and To Time fields. The refresh time is displayed below Auto Update. By clicking on Auto Update a second time, the automatic refresh is reset and the From Time and To Time fields are no longer protected. You can also deactivate Auto Update by clicking on Clear.
Update List	Click on Update List to refresh the message list.
Page Up / Page Down	Click on Page Up to go forward one page (more current), and Page Down to go backward one page (earlier).

Menu bar items:

File -	Exit	Choose **Exit** to exit the Station - System Message window
Action -	Update List	Used to refresh the system message list to show the most recent system messages.
	Reset Filter	To reset the filter to the default settings.
	Page Up	All system messages are stored in a number of buffers, each containing 100 system messages. You can change the number of buffers in the Settings menu. If you select Action - Page Up you get the previous (older messages) buffer.
	Page Down	Same as Action -Page Up, except that you get the next buffer rather than the previous one.
Help -		See Section 4.5.7 The Help Menu for a description of the menu items.

For a more detailed description of one or more system messages, select the message(s) with the mouse pointer. The Examine, Print and Save buttons in the Selection Action area on the screen become active.

When you click on **Examine** the Examine window for the selected message is opened, Figure 4-20. Click on **Print** to print the error messages or **Save** to store the message in a text file.



```
1997-08-27 14:20:42
        Message severity: Information, Message audience: Engineer
        Message: dcTTDLog has performed a normal termination
        Object:  dcTTDLog
        File:    main.C
        File Version: TTD 1.5
        Line number:  425

1997-08-27 14:27:15
        Message severity: Information, Message audience: Operator
        Message: HS:HISTORY Main History ACTIVATED
        Object:  $HSNODE_31-17
        File:    hsSrvrInit.c
        File Version: $Id: hsSrvrInit.c,v 15.1 1997/05/21 14:31:04 history $
        Line number:  708
```

Figure 4-20. System Messages Examine Window

References to OMF in system messages may include status codes. They are described in the following include file: `/opt/advant/OMF/include/omfStatus.h`.

4.4.8 How to Resolve Object References (Master)

When you choose **Object Reference Maintenance** from the **Station** menu, the Object Reference Maintenance dialog box (or main dialog box) is displayed, Figure 4-21. The Object Reference Maintenance display provides the following functions (the names of the corresponding displays):

- Retranslate Object References
- Open gateway
- Find Object
- List Object References
- Set Restart Option.

As you select functions and display sub-dialogs from the Object Reference Maintenance dialog, the main dialog box will remain on screen. Several sub-dialogs can be active at the same time.

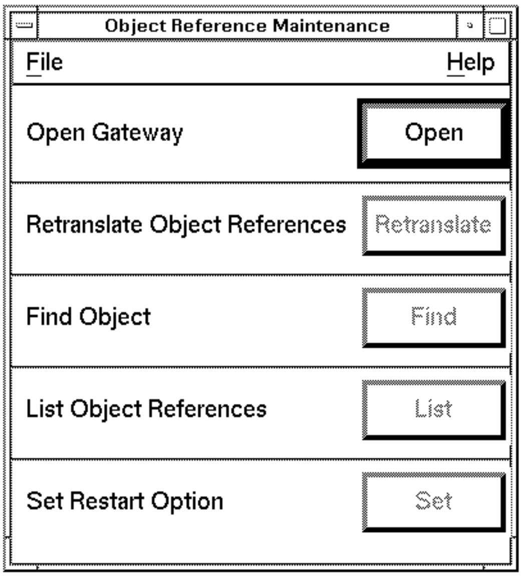
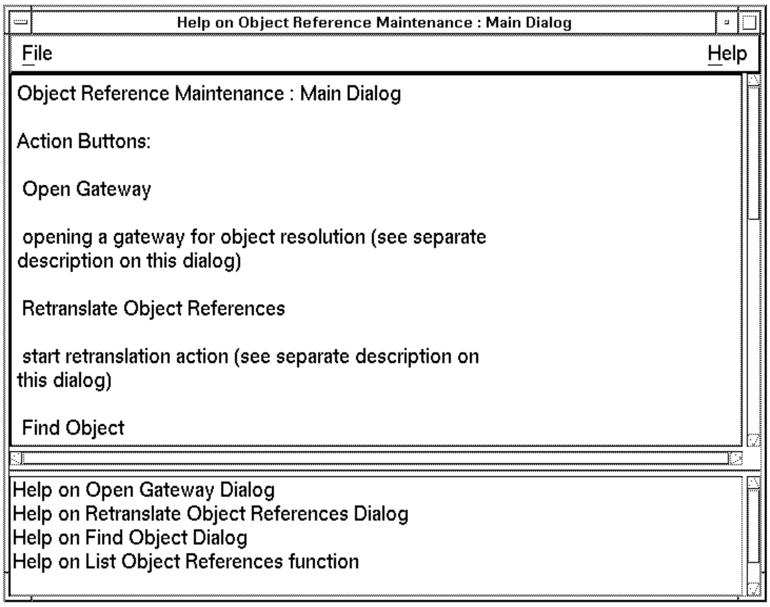


Figure 4-21. Object Reference Maintenance dialog box

Menu bar

File -	Exit	Close the Object References Maintenance dialog box. Any open subdialogs will also be closed. If you have initiated some action, the result may not be saved. However, the action itself will not be stopped. In the case of started retranslations, these actions are not stopped since they run as background jobs which do not report to this dialog when completed. In the case of listing actions and so on, which report to this dialog when completed, they will not be interrupted by a dialog exit, but the result will be discarded.
Help -	Help	Bring up the general Help dialog, with topic "Main Dialog" active - the corresponding help text is presented:



Object Reference Maintenance : Main Dialog

Action Buttons:

Open Gateway

opening a gateway for object resolution (see separate description on this dialog)

Retranslate Object References

start retranslation action (see separate description on this dialog)

Find Object

Help on Open Gateway Dialog
Help on Retranslate Object References Dialog
Help on Find Object Dialog
Help on List Object References function

Figure 4-22. Help on Object maintenance Maintenance Help information box

This window is opened whenever Help is requested (exception: when help is requested in error dialogs, a diagnosis dialog may be presented instead).

When the help dialog is requested from a help button, the correct topic is already selected. The header of the topic (the line shown in the topics list) is presented as the window title when the topic is selected.

When the help window is present, you may select any of the topics (with double-click) in the bottom of the window.

Help has a number of items in the menu bar:

- File - Close: Will close the help dialog

- File - Keep text: Will keep the current text in the help dialog window. If a new topic is selected (by double-clicking in the topics list) a new help information box is displayed.

A small toggle button is shown "pushed-in" for the Keep text function in the File menu. To turn off the Keep text option - select it again in the menu.

Buttons

Open	Open Gateway. If no gateway is connected to the current network, the Open button is dimmed, and the Open Gateway dialog is not available.
Retranslate	Retranslate Object References.
Find	Find Object.
List	List Object References.
Set	Set Restart Option.

All buttons are described in greater detail below under each sub-function.

4.4.8.1 Open Gateway

When you click on **Open** in the main dialog box, the Open Gateway dialog box is displayed, Figure 4-23.

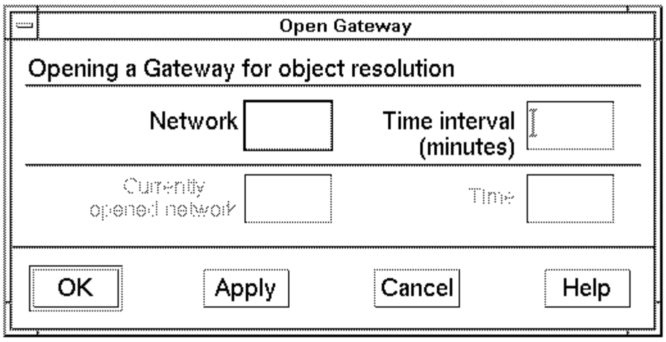


Figure 4-23. Open Gateway dialog box

This dialog box lets you open a gateway for a specified time. The purpose is to open for the retranslation and find activities and to avoid automatic and implicit opening of gateways.

Text fields

Network	The network number to which the gateway shall be opened. The following rules apply for network numbers: - The range of valid values is 11 to 127 - Wildcards are not allowed - Number must not start with '0' - Numbers ending with '0' are not allowed (denote more than one network). - An empty string is not allowed
Time	The number of minutes for which the gateway shall be kept open. The following rules apply for Time: - Range: 1 - 126 minutes. - An empty string is not allowed
Currently opened network (for presentation only)	When the dialog is presented, the number of the previously opened network gateway is shown in this field. If a new network is specified for opening (in the Network field), this previously opened gateway is automatically closed. If no gateway is currently open, this text field, including the label, is dimmed. If the network is a plant network, the network number will be set to 100.
Time (for presentation only)	The total open time for the gateway. If no gateway is currently open, this text field and label will be dimmed.

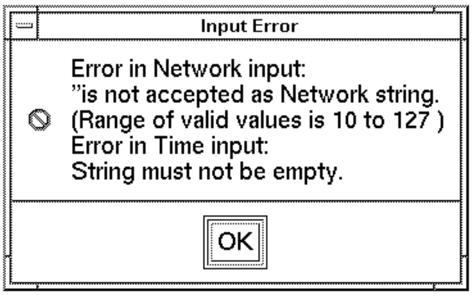
Buttons

OK

When you click on OK the currently entered data (in the Network and Time interval fields) is checked and, if no error is detected, the data is stored.

If data is accepted the opening will be performed, and the Open Gateway dialog box is closed. The only visible feedback from this operation is that the Network figure is displayed in the Currently opened network text field the next time the dialog is requested.

If entered data is not valid, the Input Error message box is displayed, Figure 4-24.



Error in Network input:
”is not accepted as Network string.
(Range of valid values is 10 to 127)
Error in Time input:
String must not be empty.

Figure 4-24. Input Error message box

OK - The Input Error information box indicates why the input was not accepted, and also which parameter (Network or Time) caused the error.

Click on OK to close the dialog box.

Apply

Input is checked and so on, according to the action for OK, but the Open Gateway dialog will remain presented (and dimmed during the open gateway call).

Cancel

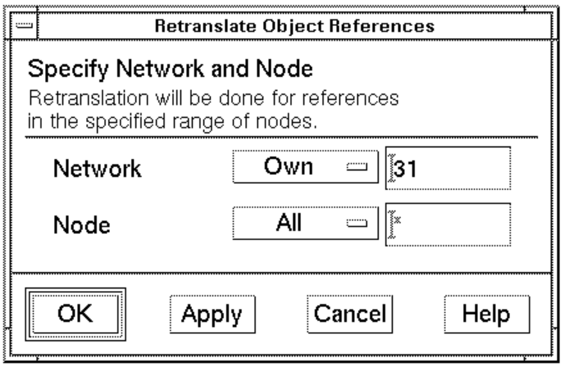
Open Gateway Dialog is closed (no open action is performed).

Help

The general Help dialog is presented, with the topic “Open Gateway” active - the corresponding help text is presented.

4.4.8.2 Retranslate Object References

When you click on the **Retranslate** button in the main dialog box, the following dialog box is displayed, Figure 4-25.



Specify Network and Node
Retranslation will be done for references
in the specified range of nodes.

Network Own 31
Node All

Figure 4-25. Retranslate Object References dialog box

This dialog lets you do retranslation for all object references to a range of nodes, specified by the Network and Node inputs.

If your own node is connected to a redundant network, you only have to give one of the network numbers. Retranslation for object references translated for the redundant network is done automatically.

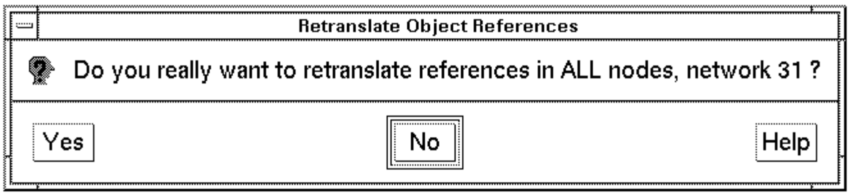
The retranslation will then be done in the range of available nodes - depending on which (if any) gateway you have opened.

Input text fields

Network	If you click on the menu next to the text field you get the following choices: - **Own** (=current network), inserts your own network in the text field and makes in unavailable. - **Specified**, makes the text field editable. The following rules apply for network numbers: - The range of valid values is 11 to 127, where any numeric character may be replaced with the wildcard (‘*’) character - Number must not start with ‘0’ - Numbers ending with ‘0’ are not allowed (denote more than one network). - An empty string is not allowed
Node	If you click on the menu next to the text field you get the following choices: - **All**, clears the text field and makes it unavailable. - **Specified**, makes the text field editable. The following rules apply for node numbers: - The range of valid values is “1” to “99” and “127”, where any of the numeric characters may be replaced with the ‘*’ character. - Numeric string must not begin with ‘0’ - An empty string is not allowed.

Buttons

OK	If the entered data is accepted (valid) a confirmation dialog box is displayed, Figure 4-26. If entered data is not accepted the Input Error information box is displayed. See the previous description under Open Gateway. A confirmation dialog box is displayed to let you verify the activation of the retranslation process.



Retranslate Object References

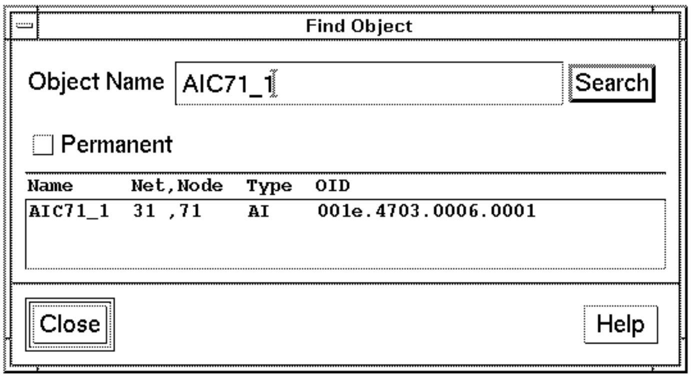
Do you really want to retranslate references in ALL nodes, network 31 ?

Figure 4-26. Retranslate Object References confirmation dialog box

Yes	Activates the translation and closes the confirm dialog as well as the Retranslate Object References dialog.
No	No action is performed, dialog is closed.
Help	Presents the general help dialog, with the topic “Retranslate Object References” active.

4.4.8.3 Find Object

When you click on the **Find** button in the main dialog box, the Find Object dialog box is displayed, Figure 4-27.



Find Object

Object Name AIC71_1

☐ Permanent

Name	Net,Node	Type	OID
AIC71_1	31 ,71	AI	001e.4703.0006.0001

Figure 4-27. Find Object dialog box

This dialog box lets you search for a specific object. The search is done within the available range of nodes (this also depends on what gateways are open).

The dialog also lets you make the found object a permanent object entry in the Name Broker, by setting the Permanent toggle button on prior to the search.

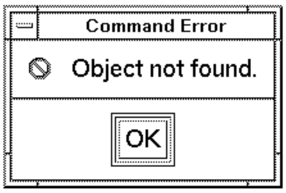
Input text fields

Object Name	Name of the object to be searched for. The following rules apply for object names: - Valid characters are: ‘0’ to ‘9’, ‘a’ to ‘z’, ‘A’ to ‘Z’, ‘-’, ‘_’, ‘,’, and ‘.’ - In addition to the above characters, the set of “national” characters is also allowed (for example å,ä and ö). - Wildcards (‘*’) are not allowed - An empty name is not allowed. If you write the object name with lowercase letters they will automatically be replaced with uppercase letters.

Output text fields

OID	Local identification of resolved object address.

Buttons

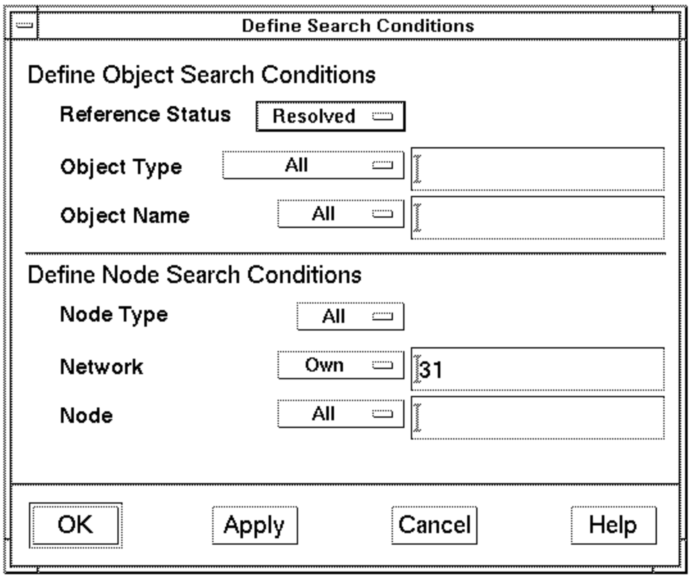
Search	When you click on Search and the entered object name is valid, a search is started. An invalid input causes the Input Error information box to be displayed. If any object is found, the object information is presented in the list below the Permanent button. If no object is found, the Find Object information box is displayed with the message: Object NN was not found, Figure 4-28. Command Error Object not found. *Figure 4-28. Find Object information box* Click on OK to close the box, or on Help to get help information.
Permanent	Toggle values: - Released: the object (if found) will be a transient object entry only, and immediately removed from the Name Broker. - Pressed: the object (if found) will become a permanent object entry in the Name Broker.

Close	Closes the Find Object dialog box.
Help	The general help is displayed, and the topic “Find Object” is active, the corresponding help text is displayed.

4.4.8.4 List Object References

Define Search Conditions

When you click on the **List** button in the main dialog box, the Define Search Conditions dialog box is displayed, Figure 4-29.



Define Search Conditions

Define Object Search Conditions

Reference Status: Resolved
Object Type: All
Object Name: All

Define Node Search Conditions

Node Type: All
Network: Own 31
Node: All

Figure 4-29. Define Search Conditions dialog box

This dialog box lets you define search conditions for the list. Once the search conditions have been entered, and the search has ended with some result, the actual list dialog is displayed.

Buttons

Reference Status

Status of the object reference in the Name Broker. There are two alternatives you can select from the menu:

- **Resolved** (default)

- **Unresolved**, causes all other options to be dimmed.

Object Type

There are three alternatives you can select from the menu:

- **All** (default)

- **Specified**, enter object type value in the text field (no wildcards allowed).

- **Select from List**, displays a subdialog containing a list of object types to select from, Figure 4-30.

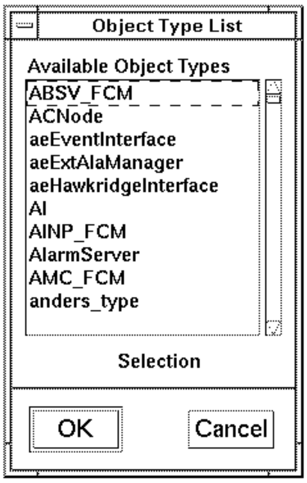


Figure 4-30. Object Type List dialog box

OK - the selected value is inserted in the Object Type text field of the Define Search Conditions dialog. Also the option case is changed from "Select from List" to "Specified".

Cancel - closes the Object Type List Dialog box and inserts no object type into the text field.

Object Name

There are two alternatives in the menu:

- **All**, default value

- **Specified**, type an object name in the text field.
Wildcard characters '?' (= any single character) and '*' (=any sequence of characters) can be used.

Node Type

There are three alternatives in the menu:

- **All** (default)
- M**aster**, means MasterPiece, MasterView, SuperView 100, SuperView 900 and Advant Controller (AC)
- **Advant**, means the Advant Station 500 Series Operator Station (AdvaCommand), Information Management Station (AdvaInform) and Engineering Station (AdvaBuild).

Network

There are two alternatives in the menu:

- **Specified**, enter a network number into the text field.
- **Own**, local.

The following rules apply for network numbers:

- The range of valid values is 11 to 127, where any of the characters may be replaced with the wildcard ('*') character
- Number must not start with '0'
- Numbers ending with '0' are not allowed (denote more than one network).
- An empty string is not allowed

Node

There are three alternatives in the menu:

- **All** (default)
- **Specified**, enter node address number in the text field.
- **Own**, local.

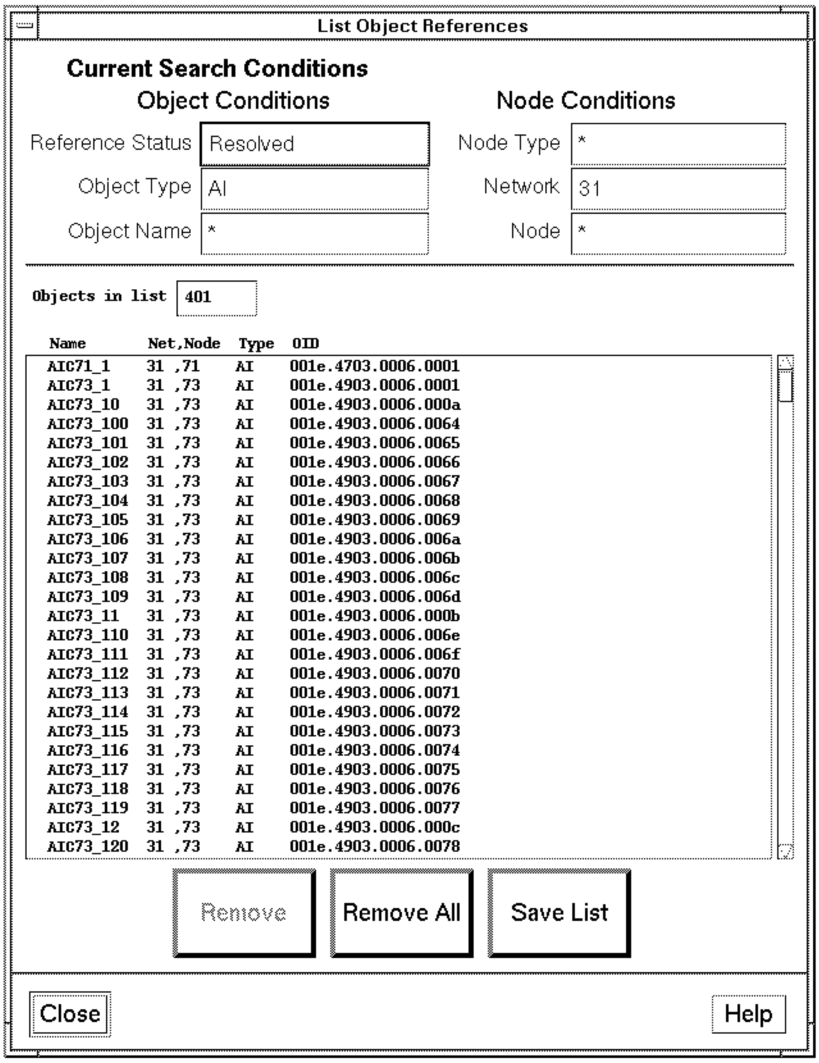
The following rules apply for node numbers:

- The range of valid values is 1 to 99 and 127, where any of the numeric characters may be replaced with the wildcard ('*') character
- Numeric string must not begin with '0'
- An empty string is not allowed.

OK	If input is valid, then start the list search, and close the Define Search Conditions dialog. If input is not valid, an Input Error Dialog is displayed to explain what parameter was not accepted and why. The Defined Search Conditions dialog will remain to enable new input values to be entered. When the search starts, the Reference List dialog box is displayed (dimmed and empty) and the cursor is displayed as a watch. When the search is completed, the watch cursor returns to its normal shape, and the resulting list of references is presented in the Reference List dialog box. If the search failed - that is, no references matched the defined search conditions - an error information box is displayed.
Apply	If input is valid, start the list search, and continue to display the Define Search Conditions dialog box. Otherwise, see OK above.
Cancel	Close the Define Search Conditions dialog box.
Help	The general help dialog is displayed with the “Define Search Conditions” topic active.

Reference List

When you activate the list search by pressing **OK** or **Apply** in the Define Search Conditions dialog box and the search concludes with some result, the following dialog box is displayed, Figure 4-31.



Current Search Conditions

Object Conditions

Reference Status: Resolved
Object Type: AI
Object Name: *

Node Conditions

Node Type: *
Network: 31
Node: *

Objects in list: 401

Name	Net,Node	Type	OID
AIC71_1	31 ,71	AI	001e.4703.0006.0001
AIC73_1	31 ,73	AI	001e.4903.0006.0001
AIC73_10	31 ,73	AI	001e.4903.0006.000a
AIC73_100	31 ,73	AI	001e.4903.0006.0064
AIC73_101	31 ,73	AI	001e.4903.0006.0065
AIC73_102	31 ,73	AI	001e.4903.0006.0066
AIC73_103	31 ,73	AI	001e.4903.0006.0067
AIC73_104	31 ,73	AI	001e.4903.0006.0068
AIC73_105	31 ,73	AI	001e.4903.0006.0069
AIC73_106	31 ,73	AI	001e.4903.0006.006a
AIC73_107	31 ,73	AI	001e.4903.0006.006b
AIC73_108	31 ,73	AI	001e.4903.0006.006c
AIC73_109	31 ,73	AI	001e.4903.0006.006d
AIC73_11	31 ,73	AI	001e.4903.0006.000b
AIC73_110	31 ,73	AI	001e.4903.0006.006e
AIC73_111	31 ,73	AI	001e.4903.0006.006f
AIC73_112	31 ,73	AI	001e.4903.0006.0070
AIC73_113	31 ,73	AI	001e.4903.0006.0071
AIC73_114	31 ,73	AI	001e.4903.0006.0072
AIC73_115	31 ,73	AI	001e.4903.0006.0073
AIC73_116	31 ,73	AI	001e.4903.0006.0074
AIC73_117	31 ,73	AI	001e.4903.0006.0075
AIC73_118	31 ,73	AI	001e.4903.0006.0076
AIC73_119	31 ,73	AI	001e.4903.0006.0077
AIC73_12	31 ,73	AI	001e.4903.0006.000c
AIC73_120	31 ,73	AI	001e.4903.0006.0078

Figure 4-31. List Object References dialog box

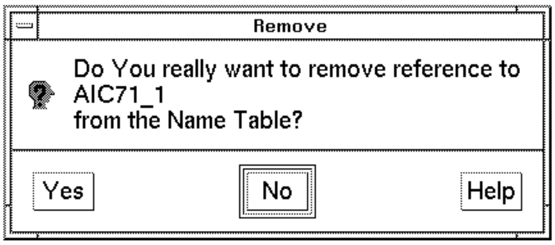
The dialog box has two presentation parts:

- Current Search Conditions (top of box)
- List of references - This is the result from the previous search.

Buttons

Remove

Removes the references from the Name Broker. A confirm dialog, Figure 4-32, is displayed when the **Remove** button is pressed.



Do You really want to remove reference to
AIC71_1
from the Name Table?

Figure 4-32. Remove confirmation dialog box

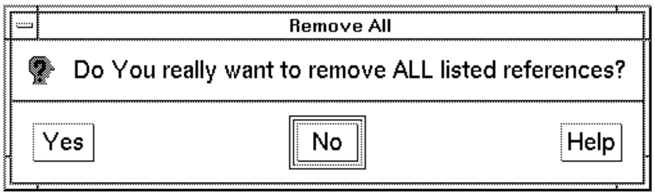
This action is only available when one or more removable references are selected in the list.

Note:
If a reference which is not allowed for removal is selected within a multiple selection, the remove action will not have any effect on this reference.

When a reference has been removed, it is marked in the Reference List Dialog with a "v-" in front of the object name.

Remove All

Removes all listed references from the Name Broker. A confirm dialog, Figure 4-33, is displayed when the **Remove All** button is pressed.



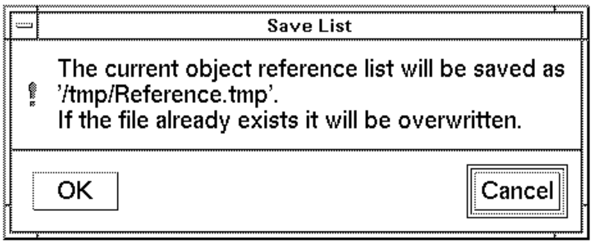
Do You really want to remove ALL listed references?

Figure 4-33. Remove ALL confirmation dialog box

Removed references are marked in the Reference List Dialog with a “v-” in front of the object name. The Remove All action causes all references to be marked - except for those that were not allowed to be removed.

Save List

Saves the current list on the file `/tmp/References.tmp`. The same file name is always used, and previous versions of this file will be overwritten, so when you click on the **Save List** button the Save List dialog with a warning is displayed, Figure 4-34.


Save List

The current object reference list will be saved as '/tmp/Reference.tmp'.
If the file already exists it will be overwritten.

Figure 4-34. Save List warning dialog box

The Save List warning dialog has the following buttons:

OK - saves the list to the specified file and the warning dialog is closed.

Cancel **-** no save is done and the warning dialog is closed.

Help

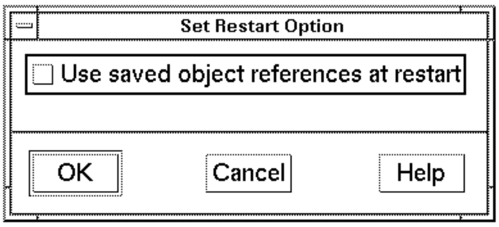
Displays the general help dialog, with topic “List Object References” active and the corresponding help text presented.

Close

Closes the Reference List dialog.

4.4.8.5 Set Restart Option

When you click on the **Set** button in the main dialog box, the following dialog box is displayed, Figure 4-35.


Set Restart Option

☐ Use saved object references at restart

Figure 4-35. Set Restart Option dialog box

This dialog box lets you define the restart behavior of the station concerning the usage of translated and stored object references. If you specify that the stored object references shall be used, no translation for those references will be done in the connected network. This means that all applications (History, User Objects and so on) will start up without name resolution.

NOTE

Since name resolution will start up as a background job if objects are not resolved, it may take some time until all symbols have been resolved. For that reason, the default setting is to use saved object references at restart.

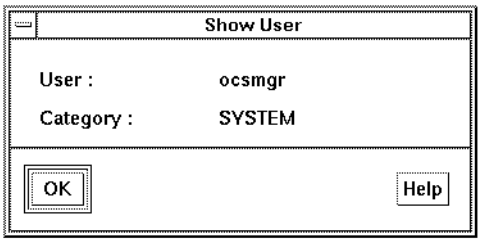
When this dialog box is displayed, the toggle button shows the current value of this option.

Buttons

OK	The dialog is closed, and a call is made to set the restart option according to the entered value on the toggle button.
Cancel	The dialog is closed, any change in the option is discarded.
Help	The general help dialog is displayed, with the topic “Set Restart Option” active: the corresponding help text presented.

4.4.9 How to Show Current User

When you choose **Show User** from the **Session** menu, the current user dialog is displayed, Figure 4-36. This dialog indicates the User Name and Category.



User : ocsmgr

Category : SYSTEM

Figure 4-36. Show User window

These fields are described below:

User	The user name used at login to AdvaInform
Category	The user categories are: System, Engineer and Operator. For further information refer to Section 3.3.1, User Authorization.

4.4.10 How to Change Language

There are two ways to get native language support when you use AdvaInform:

- All AdvaInform user's have a default language set-up when the user is defined. When you log in the language is set up automatically. See the *Advant Station 500 Series IMS User's Guide* for more information how to define the default language.
- You can temporarily change your language. This is described below.

Changing the language

To temporarily change the language for your AdvaInform user, choose **Language** from the **Session** menu and select the desired language, Figure 4-37.

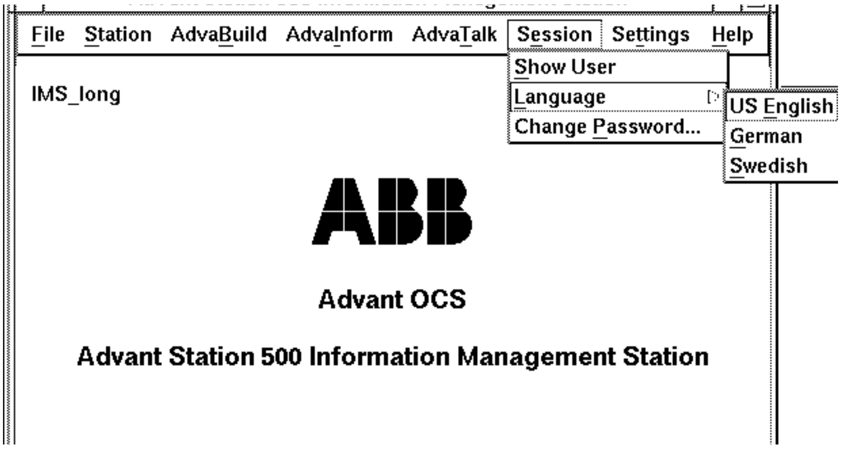


Figure 4-37. Session menu- Language

CAUTION

When you switch language all windows opened from the AdvaInform menu that you are currently executing are closed. The AdvaInform menu is restarted with the new language set-up. Thus you must make sure that is it OK to close all the windows before you switch language.

This does not affect other programs or users currently logged in. To make the new language be in effect for your previously started terminal windows, editors and so on, you must restart them.

As a result of the language switch, the environment variables for native language support will all be set to the new language. Thus, all HP tools, Oracle forms and so on, that support the new language will also use the new language.

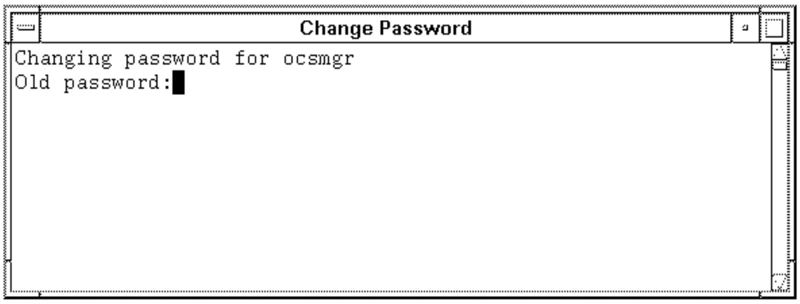
When you click **OK** in the Set Language window, the environment variables LANG and NLS_LANG are set to the selected language and the AdvaInform Menu is restarted. This means that all open AdvaInform menu windows that were closed are opened from the menu, and a new AdvaInform menu is started with the selected language.

NOTE

You must restart AdvaBuild-, AdvaInform and AdvaTalk-options, Oracle forms and terminal windows to get correct language in those applications.

4.4.11 How to Change Password

When you choose **Password** from the **Session** menu, the Change Password window is displayed, Figure 4-38.



```
Changing password for ocsmgr
Old password:█
```

Figure 4-38. Change Password window

Enter your old password then press <**Return**>. You will then be requested to enter a new password and to confirm it.

4.5 Runtime Menus

4.5.1 File Menu

The **File** menu has just one menu item: **Exit**.

4.5.1.1 File - Exit

Use **Exit** to close the IMS Menu.

4.5.2 Station Menu

The **Station** menu is shown in Figure 4-39.

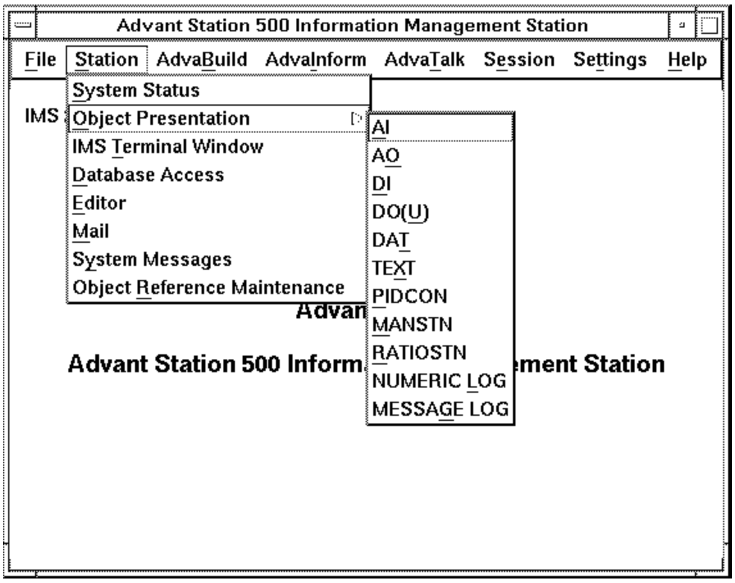


Figure 4-39. Station menu (Master)

4.5.2.1 Station - System Status

The System Status differs between Master and MOD 300. For systems with MOD 300 software, it consists only of the Node Status Display. In systems with Master software, the complete functionality is supported as described in Section 4.4.3, How to Use System Status.

4.5.2.2 Station - Object Presentations

Use this menu item to open Object forms. When you choose **Object Presentations**, a submenu with the available forms displayed, Figure 4-40 (for Master), The available forms for systems with Master software are: AI, AO, DI, DO, DAT, TEXT, PIDCON, MANSTN, RATIOSTN, NUMERIC LOG, and MESSAGE LOG..

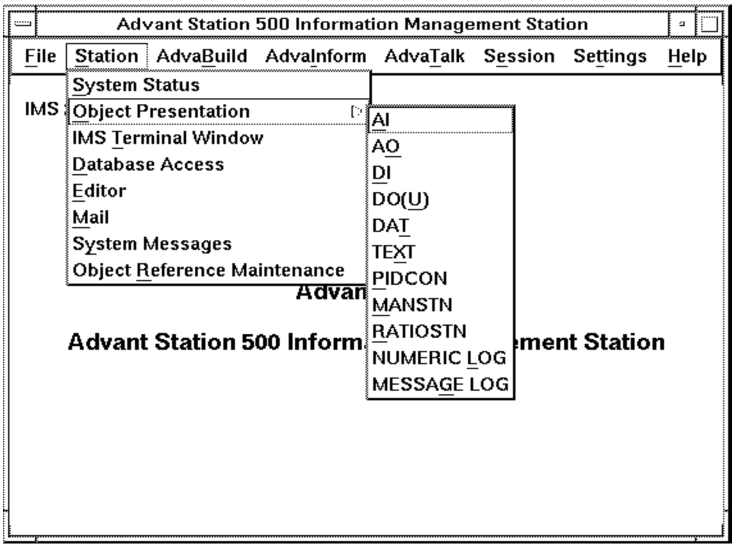


Figure 4-40. Object Presentation - Object Type Menus (Master)

The available forms for systems with MOD 300 software are: AI, AO, DI, DO, DAT, TEXT, NUMERIC LOG, MESSAGE LOG, and MOD 300 MESSAGE LOG.

Choose the applicable object presentation from this menu to open the Object Presentation. Specific instructions for using these forms are provided in Section 4.4.2.6, How to Present Objects and Logs.

4.5.2.3 Station - Terminal Window

Use this menu item to open a new terminal window with a command line prompt. The prompt is ksh (*$* by default) for Advant Station 500 Series. The window is an ordinary terminal window that provides a specific AdvaInform environment. You need this environment to run the AdvaInform standard tools, or your own AdvaInform application programs.

To exit or close the terminal window, use the window pull-down menu in the upper, left corner and then choose the **Close** option. Another way to close the window is to log out.
In Advant Station 500 series, enter the command **exit**.

4.5.2.4 Station - Database Access

Use this menu item to open a terminal window and start SQL*Plus. You are automatically logged in as your Oracle user. From this window you can use SQL and PL/SQL queries into database tables and views.

4.5.2.5 Station - Editor

Use this menu item to start your default editor.

4.5.2.6 Station - Mail

Use this menu item to start the default electronic Mail system.

4.5.2.7 Station - System Messages

The System Messages window displays messages from the basic software as well as from the AdvaInform, AdvaTalk and AdvaBuild options. By using an interface in the AdvaInform User API, you can also send messages to the log from your own application programs.

The System Message window has features to selectively display the current logs in the system, and to control the presentation in other ways. The window displays system messages that occur between a **From Time** and a **To Time**, and that satisfy the **Severity** and **Audience** qualifiers.

4.5.2.8 Station - Object Reference Maintenance (Master only)

The Object Reference Maintenance display provides functions to maintain and resolve object references.

For simple configuration and fast addressing of the different objects of the Advant OCS database, AdvaInform automatically translates referred object names into object references. This information is stored in Core system.

Changes in the Advant OCS database, with respect to the information stored in the Name Broker, which occur *after* the original commissioning cannot be detected by the system. An example of such a change is when an object is moved within the database and is assigned a new address. The translation (retranslation) of the object references must then be initiated manually.

You can request retranslation to objects in all nodes within your own network, to objects in a specified node, or to objects in all nodes connected to a specified network. The network can be either a control network or a plant network. You can also check the reference for a specified object.

Information about resolved object references is automatically stored. This information is normally used each time an object is accessed by any AdvaInform function or application.

All objects are located in nodes - process stations, operator stations, and IMS. The address of a node is the combination of a network number and a node number within the network. When you request a retranslation of references to objects in a network - own or a specified control/plant network - the references to objects in all nodes within the specified network are retranslated.

4.5.3 AdvaBuild Menu

Use this menu to start any of the AdvaBuild options installed on your IMS, Figure 4-41. Only installed options are shown on the menu. If an option is installed but you do not have the authority to use it, the option name is dimmed.When new options are installed they are automatically added to the menu. For details about a specific AdvaBuild option, consult the corresponding User's Guide

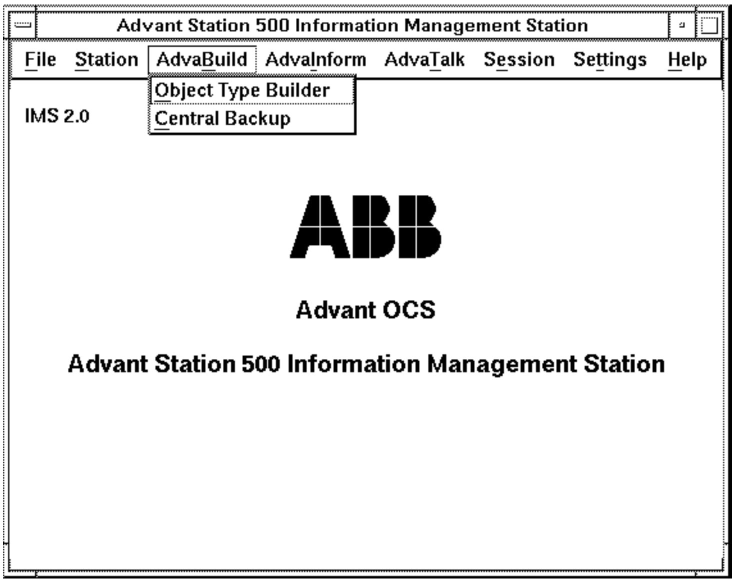


Figure 4-41. AdvaBuild menu

4.5.3.1 AdvaBuild - Object Type Builder

When you choose Object Type Builder from the AdvaBuild menu a terminal window is opened. From this window you can enter commands to start the different tools for building User Object types. For more information see the *AdvaBuild Object Type Builder User´s Guide*.

4.5.3.2 AdvaBuild - Central Backup (Master only)

The Central Backup utility is used for handling of backups of applications in. Central Backup can be used for AC410/450, MP200/1, MV800/1 and MG 230/1 nodes. For more information see the *AdvaBuild Central Backup User´s Guide*.

4.5.3.3 Programmer 2000 - Pro*C

The Programmer 2000 - Pro*C option provides a pre-compiler to embedded SQL and PL/SQL. This means that you can develop application programs accessing the relational database and all the AdvaInform SQL*Connect and SQL*Connect Programming functions.

4.5.4 AdvaInform Menu

Use this menu to start any of the AdvaInform options installed on your IMS, Figure 4-42. Only installed options are shown on the menu. If an option is installed but you do not have the authority to use it, the option name is dimmed. When new options are installed they are automatically added to the menu. For details about a specific AdvaInform option, consult the corresponding User's Guide.

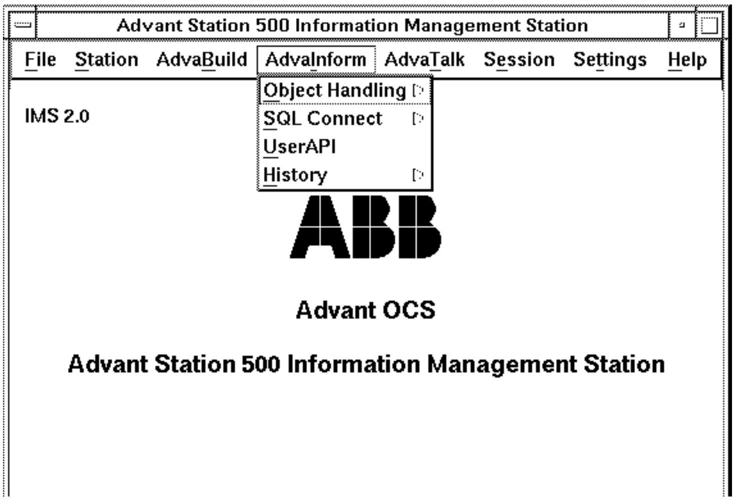


Figure 4-42. IMS menu example for Master

4.5.4.1 AdvaInform - Object Handling

Choosing **Object Handling** displays a cascade menu with the following menu items:

- **Object Configuration**: Provides configuration facilities for AdvaInform Basic and User objects.
- **Object Handler**: Provides access to the object handling utilities.
- **Object Status:** Provides an on-line overview for the status of all objects.

For more information see the *AdvaInform Object Handling User´s Guide*.

4.5.4.2 AdvaInform - SQL*Connect

This menu item provides access to the configuration tools for AdvaInform SQL*Connect and displays a cascade menu with the following menu items:

- **Client Object Instance Configuration**: To add object instances to the Location Table.
- **Client Object Type Configuration**: To add and configure supported Object Types.
- **Server Configuration**: To connect data sources (tables or views) to Object Types.

Refer to the *AdvaInform SQL*Connect User's Guide* for details.

4.5.4.3 AdvaInform - User API

This menu item opens a terminal window. The window provides guidelines on how to find API examples. In this environment you can compile your user API applications. For more information see the *AdvaInform User API User´s Guide*.

4.5.4.4 AdvaInform - History

This displays a cascade menu with the following menu items:

- **History Configuration:** Provides access to the AdvaInform History option which lets you configure logs, log groups, log sets and so on.
- **TTD Configuration:** Starts a tool which allows you to configure TTD-logs towards AdvaInform History.
- **History status**: Displays History Runtime Status display to view the state of History logs.

Refer to the *AdvaInform History User’s Guide* for details.

4.5.4.5 AdvaInform - Calculations (Master only)

When you select AdvaInform Calculations, another menu is displayed:

- **Calculation Builder**: To create calculation modules and instances.
- **Calculation Status**: Provides an overview of all calculations. Calculations can also be executed using this tool.

For more information see the *AdvaInform Calculations User´s Guide*.

4.5.4.6 AdvaInform - Reports

This displays a cascade menu with the following menu items:

- **Reports Configuration**: A tool for building report templates
- **Manual Execution**: Use this to execute a report manually.
- **Reports Status**: Provides an overview of all your reports, and lets you activate reports.
- **View Logs**: Provides access to Reports stored in History.

Refer to the *AdvaInform Reports User’s Guide* for details.

4.5.4.7 AdvaInform - Event Log (Master only)

This displays a cascade menu with the following menu items:

- **Event Log Configuration**: This opens the Event Log Configuration window for configuration and handling of subscription on events stored in MasterView and AdvaCommand. It is basically the same function as in SuperView 900.
- **Event Log Status**: Provides an overview of all your event logs.

For more information see the *AdvaInform Event Log User´s Guide*.

4.5.4.8 AdvaInform - SPC (MOD 300)

This opens the SPC (Statistical Process Control) Configuration window. Refer to the *AdvaInform SPC User's Guide* for details.

4.5.4.9 AdvaInform - PDL (MOD 300)

This menu item opens the PDL (Production Data Log) window. Refer to the *AdvaInform PDL User's Guide* for details.

4.5.5 AdvaTalk Menu

AdvaTalk - Link to MATLAB

This displays a cascade menu with the following menu items:

- **Configuration**: To configure run time object instances.
- **MATLAB**:

For more information see the *AdvaTalk Link to MATLAB User´s Guide*.

4.5.6 Session Menu

The Session menu provides access to information about the current user, and lets you change language and password, Figure 4-43.

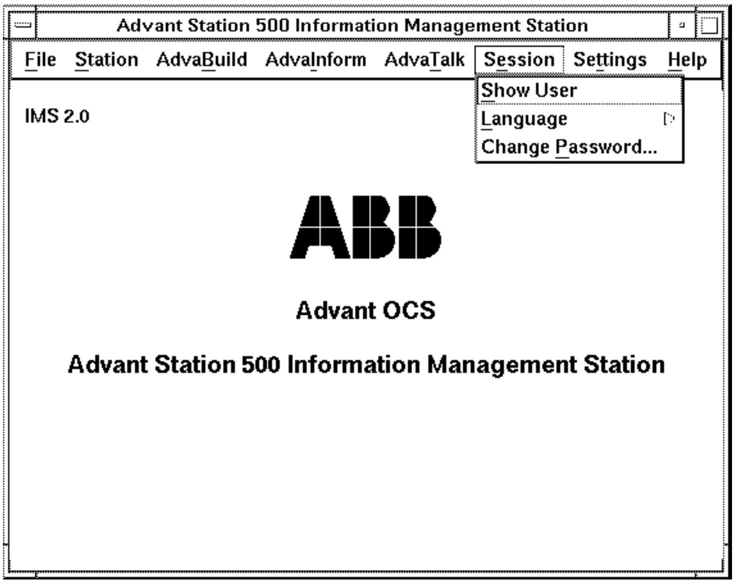


Figure 4-43. Session Menu

Session - Show User

Use this menu item to show current user and user category.

Session - Language

The standard AdvaInform includes language support for the following languages:

- US English (complete support)
- Swedish, German
  - support for keyboards with native language characters
  - support for the native language in HP tools such as editors
  - support for all displays available for the user category operator.

In addition your country's local ABB representatives can translate the AdvaInform user interface to give you the same level of language support as described for German and Swedish above.

Session - Password

Use this menu item to change your current password.

4.5.7 Help Menu

The **Help** menu, Figure 4-44, provides help functions related to the AdvaInform software functions.

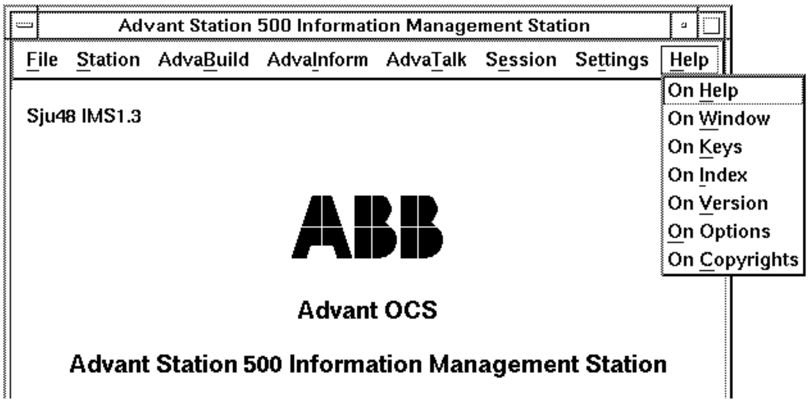


Figure 4-44. AdvaInform Help Menu

Whenever you need information about your current display, you can select related items on the Help menu. You can find general information about operation of OSF/Motif displays as well as specific information about certain displays. Help is always presented as a Help Window with help text.

Help - On Help

Provides information about how to use the Help menu itself.

Help - On Window

Describes how to operate the current window.

Help - On Keys

Provides information about key operation, for example how to use Accelerator keys.

Help - On Index

Displays a list of keywords (topics) you can get information for. Double-click on any keyword to get information.

Help - On Version

Gives information about version codes for certain software in your IMS, for example, the AdvaInform Basic Functions.

Help - On Options

Get information on the software options available.

Help - On Copyrights

A list of applicable copyrights.

Each of the help windows, accessible from the Help menu (above), has the following items on their menu bars:

File -	Keep Text	If you select Keep Text, subsequent help windows will not overlap the already displayed one(s)
	Close	Closes the help window.
Help		Gives information about the current window.

Chapter 5 Maintenance

5.1 Preventive Maintenance

Refer to Chapter 5 in the *Advant Station 500 Series IMS User’s Guide*.

5.2 Hardware Indicators

Hardware indicators are described in the *Advant Station 5XX Hardware User’s Guide*.

5.3 Presenting System Messages

System messages generated by AdvaInform Basic Functions and options can be viewed in different ways. For example you can choose to display only those having a certain severity code by using the features of the System Messages display (described in Chapter 4), or by using the bsLogTool as described in the *Advant Station 500 IMS User´s Guide*.

The system messages issued by AdvaInform Basic Functions and options have the following standardized format:

Error string <P1><P2><P3>

Parameters:	P1: description of parameter 1 P2: description of parameter 2 P3: description of parameter 3 The parameters are optional in a system message. This means that there might be no parameter, just text, or one, two or three parameters.
Audience:	The audience of the system message. Valid entries are Operator, Engineer, Service and Debug.
Cause:	Cause of the error.
Action:	Action to take when the error occurs.

The system messages are sorted according to their severity in the following classes

- Fatal system messages
- System messages
- Warning messages
- Information messages.

5.3.1 Oracle *oerr* on-line tool

Error messages related to *Oracle* tools can be investigated further by using the *oerr* on-line tool from a terminal window. This applies to all of the error messages issued by any of the installed Oracle tools. These types of messages are presented in two ways:

- In interactive mode (for example when using the Object Presentations) the message (starting with the ORA number) is displayed at the bottom line of the window
- In a *trace* file. This occurs when a failure in hardware or software (system or an Oracle tool) prevents an instance from continuing work.

The *trace* files reside in the directory `/products/log/oracle`. The files are in ascii format. To find the trace file of interest for your problem, check the date of the files. Type the file to get the error number and use *oerr* to get a description.

Example:

To get a better description of the Oracle message ORA-6100 than what is supplied by the message itself:

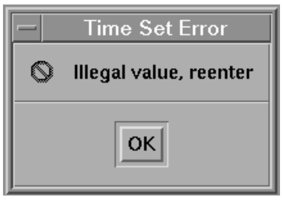
1. Open a terminal window.
2. Enter command: **`oerr ora 6100`**

This provides the cause of the error as well as a suggestion on what action you should take to fix the problem.

5.3.2 Message Box

The other type of error (or informational) messages that are issued by AdvaInform Basic Functions and options are presented directly to you. Normally, the reason for the message is that you have entered an invalid command. In most cases, you can take the necessary action to correct the command.

These types of messages are presented in a standardized message box, Figure 5-1.



Time Set Error

Illegal value, reenter

Figure 5-1. Example of an Error Message dialog box

The error message must be acknowledged (click on **OK**) to continue. The error messages are self explanatory and are not described further here.

Messages can also be presented in a scrolled message list.

5.3.3 IMS Menu

Advant Station connected to the TCP/IP network

If the error message "Can't open display" is displayed when trying to open the IMS Menu, the most likely problem is related to TCP/IP. If the TCP/IP communication was not running correctly when the software was started you will have a problem when you try to open the IMS Menu.

Check that the physical TCP/IP connection is OK. Then restart the Advant Station using the `IMSStart` command.

Advant Station not connected to TCP/IP network

If you are running the Advant Station without connecting it to the TCP/IP network, you should terminate the port. If you do not terminate the port the error message "Can't open display" is displayed when you try to open the IMS Menu. To circumvent the problem you can enter the system command:

```
$ export DISPLAY=:0
```

then try the `IMSMenu` command once more.

5.4 System Messages

5.4.1 Alarm and Event Distributor - Process bcaedistr

5.4.1.1 Fatal error messages

Failed open connection to OMF, (call = <P1>, errno = <P2>)

Parameters:	P1: Name of OMF call P2: OMF return status
Audience:	Engineer
Cause:	An error occurred in the start-up phase of the alarm and event distributor. The connection to OMF didn't succeed.
Action:	Check for error logs from OMF and if the omfPM process is running. If not running, restart IMS.

Failed to define exit handler, (call = <P1>, errno = <P2>)

Parameters:	P1: Name of c call P2: Operating system error number
Audience:	Engineer
Cause:	The alarm and event distributor failed to define an exit handler during the start-up phase. Operating system related error.

Action: Check the meaning of this error number for your operating system or contact ABB.

Failed to register A/E distributor, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer, Service

Cause: The alarm and event distributor failed to register itself in OMF

Action: Check for other error logs from OMF regarding the omfNameProc process, name broker/server, or the omfPM process. Restart IMS

5.4.1.2 Error messages

Failed to create an event, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: The alarm and event distributor failed to indicate the occurrence of an event by activating OMF. This means that local subscribers of alarms and events do not receive updates of alarms and events.

Action: Check for other error logs from OMF regarding the omfPM process. Restart IMS.

Failed to send a message to history message log (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: The alarm and event distributor failed to send an alarm or event message to the history message log.

Action: Check for other error logs from OMF regarding the omfPM process. Restart IMS.

Failed to send a message to DCS, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: The alarm and event distributor failed to distribute an alarm or event message to the nodes in the DCS subscribing for alarm and event messages.

Action: This problem can be caused by some malfunction in the OMF or by problems in the communication software connecting AdvaInform to the DCS. Check for other error logs from OMF regarding the omfPM process. Restart IMS.

5.4.1.3 Warning messages

A/E distributor not registered for supervision, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer, Service

Cause: The alarm and event distributor failed to register itself for supervision of process state. This doesn't affect the alarm and event distributor in any vital matter but may signal problems in other parts of AdvaInform Basic Functions.

Action: Check if the process dcSystatServ is running. If not, check for error messages from that process, correct and restart IMS.

No connection to history message log, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer, Service

Cause: The alarm and event distributor failed to find a history message log object to send alarm and event messages to. This can be caused by problems in the History option or if the History option is not installed. If the History option is not installed this warning is written every time the IMS is restarted.

Action: If History option is installed, check for error messages from History software.

No connection established with DCS, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer, Service

Cause: Alarm and event distributor failed to find a communication interface object to send alarm and event messages that should be distributed to the DCS.

Action: This problem can be caused by some malfunction in the OMF or by problems in the communication software connecting the AdvaInform Basic Functions to the DCS. Check for other error logs from OMF regarding the omfNameProc process. Use the MA215 utility problem to examine is there is any problem with the real time accelerator board. Restart IMS.

Connection to OMF not properly closed (errno = <P1>)

Parameters:	P1: OMF return status
Audience:	Engineer
Cause:	When the alarm and event distributor process was shut down it was not able to remove all connections to OMF. This might eventually lead to a shortage of OMF connections and an OMF crash.
Action:	No action.

5.4.1.4 Information messages

Failed to store context, (call = <P1>, errno = <P2>)

Parameters:	P1: Name of OMF call P2: OMF return status
Audience:	Service
Cause:	The alarm and event distributor object was not able to store a description of itself on disk. This information is used when IMS is restarted to recreate the alarm and event distributor with the same object identity. If the information is missing the alarm and event distributor is created with a new object identity.
Action:	No action.

Failed to terminate monitor, (call = <P1>, errno = <P2>)

Parameters:	P1: Name of OMF call P2: OMF return status
Audience:	Service
Cause:	The alarm and event distributor did not de-allocate a monitor that it had allocated. Might cause a shortage of monitors after a large number of unsuccessful de-allocations.
Action:	No action.

Failed to register A/E distributor
A/E distributor not registered for supervision
No connection to history message log
No connection established with DCS

Previously described with higher severity and audience: Service

5.4.2 System Status display (Master)

5.4.2.1 Fatal error messages

Failed open connection to OMF, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: An error occurred in the start-up phase of the system status display.
The connection to OMF did not succeed.

Action: Check for error logs from OMF and if the omfPM process is running.
Restart IMS.

Failed to define exit handler, (call = <P1>, errno = <P2>)

Parameters: P1: Name of c call
P2: Operating system error number

Audience: Engineer

Cause: The System Status display failed to define an exit handler during the start-up phase. Operating system related error.

Action: Check the meaning of this error number for your operating system or contact ABB.

ORACLE error : <P1>

Parameters: P1: ORACLE error message

Audience: Engineer

Cause: An error occurred during a Oracle access.

Action: Check the meaning of this error number with the *oerr* command.

5.4.2.2 Warning messages

Connection to OMF not properly closed, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: When the System Status display process was shut down it was not able to remove all connections to OMF. This might eventually lead to a shortage of OMF connections and an OMF crash.

Action: No action. Continue execution of IMS. Restart IMS if there is a crash.

Subscriptions to node not properly closed, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: When the System Status display process was shut down it was not able to remove all subscriptions to other nodes. This might eventually lead to a shortage of OMF connections and a OMF crash.

Action: No action. Continue execution of IMS. Restart IMS if there is a crash.

Failed init node status request, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: An attempt to access node status for a Advant node didn't succeed. There may be a number of reasons. Check that the OMF processes and the dcSysStatServ process are running.

Action: Restart IMS.

5.4.2.3 Information messages

Failed to terminate monitor, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: The System Status display was not able to close a monitor that it has opened towards OMF. This might eventually lead to a shortage of OMF monitors and a OMF crash.

Action: No action. Continue execution of IMS. Restart IMS if there is a crash.

Failed to init node status request, (call = <P1>, errno = <P2>)

Parameters: P1: Name of OMF call
P2: OMF return status

Audience: Engineer

Cause: An attempt to read node status for a Advant node did not succeed. No data was available.

Action: Check that the OMF processes and the dcSysStatServ process are running. If processes are missing, restart IMS.

Failed to init network status request, (call = <P1>, errno = <P2>)

Parameters:	P1: Name of OMF call P2: OMF return status
Audience:	
Cause:	An attempt to access network status for a node didn't succeed. There may be a number of reasons, processes missing etc.
Action:	Check that the OMF processes in both the local and the accessed node are working. If the selected node is an Advant node also check that the dcSysStatServ process are running. If processes are missing, restart IMS.

Failed read network data, (call = <P1>, errno = <P2>)

Parameters:	P1: Name of OMF call P2: OMF return status
Audience:	Engineer
Cause:	An attempt to read network status for a node didn't succeed. No data was available.
Action:	Check that the OMF processes are running. If the accessed node is a Advant node check that it has OMF processes and the dcSysStatServ process running. If processes are missing, restart IMS.

Failed translate node, (call = <P1>, errno = <P2>)

Parameters:	P1: Name of OMF call P2: OMF return status
Audience:	Engineer
Cause:	An attempt to translate names of one or more nodes failed.
Action:	Check that the OMF processes are running. If processes are missing, restart IMS.

5.5 Fault Finding and Repair

Refer to Chapter 5 in the *Advant Station 500 Series IMS User's Guide*.

Chapter 6 FAQ (Frequently Asked Questions)

How do I connect a PC to AdvaInform for data collection using SQL? Which software packages should I use? Where do I buy the packages?

SQL*Net is needed for SQL access. SQL*Net version 1.x and 2.x are provided in the Advant Station 500 IMS. SQL*Net and ODBC options for the PC are available in the Advant price list.

See the Release Notes for the SQL*NET software for installation and test of the connection.

Where should I put my custom applications?

You should place your applications under **`/opt/advant/customer/.`**

We have installed more primary memory in our IMS, but the performance seems to be worse rather than better?!

The total SWAP space in the computer must be adjusted to the size of the primary memory. See *Advant Station 500 Series IMS User´s Guide*.

Can I read TTD data without AdvaInform History installed?

Yes, you only need to map the TTD logs. Then they will be accessible both from SQL*Connect and from User API programs.

How can I communicate between IMS 500 and SuperView 900? (Master)

Cyclic subscriptions and Event subscriptions towards SuperView 900 objects require the SuperView 900 version 4.1/5. Demand subscription (Corresponds to Object Access Service: bciGetAttributes()) works towards all SuperView 900 4.1 revisions.

If you intend to access objects from SuperView 900, for instance with AdvaInform SQL*Connect - Client, lower case letters in object names are not supported by the Core System. This means that it is not possible to access objects with lower case letters in the object names.

History in SuperView 900 4.1/6 supports the functionality of the TTD mapping tool. If this functionality is required you should upgrade your SuperView 900 to 4.1/6. This makes it possible to display SV logs on Advant Station 500 OS without using IMS 500. If you want SV to subscribe on IMS 500 objects and to send order from SV to IMS 500 objects SV900 version 4.1/6 is needed.

Appendix A Relational Databases, SQL, and SQL*Plus

A.1 Relational Databases

A database management system allows you to store, retrieve and manipulate information. Currently there are three types of database management systems: Hierarchical, Network, and Relational. A brief comparison of all three systems is provided below. A more detailed description of the AdvaInform relational database follows.

A hierarchical database management system represents the data as tree structures, composed of a hierarchy of data records, Figure A-1 (compare with a file system).

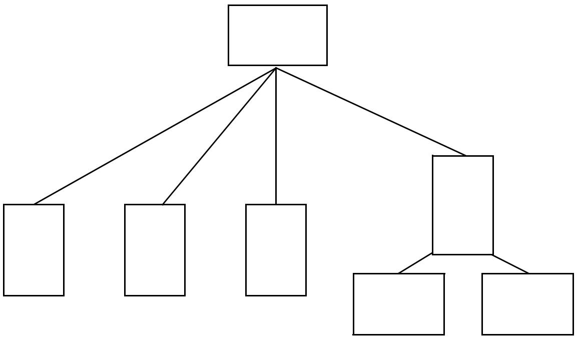


Figure A-1. Hierachical Database Model

A network database management system represents data as records linked together, forming intersects sets of data, Figure A-2.

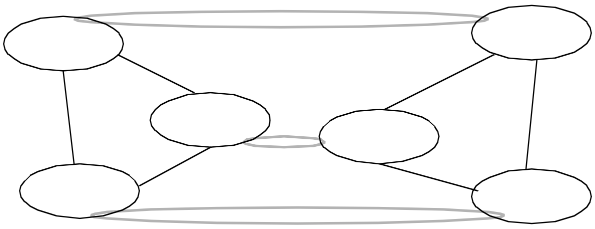


Figure A-2. A Network Database Model

A relational database relies on the fact that even the most complex hierarchical or network database can be represented as a simple collection of two-dimensional tables. This collection of tables is known as a relational database management system, Figure A-3.

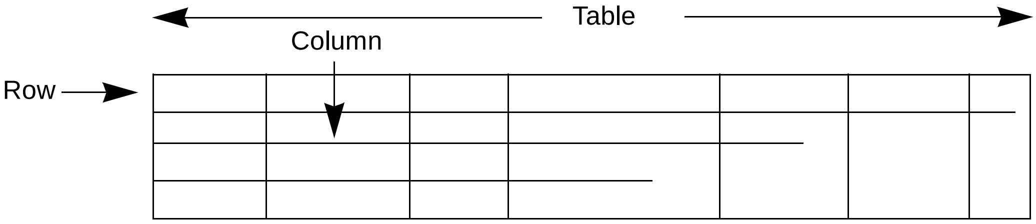


Figure A-3. Relational Database Model

In a relational database there is only one type of data structure: **tables**.

A relational database language produces new tables by combining and/or subsetting existing tables. By a single command, entire sets of records (rows) can be retrieved from one or more existing tables and stored into new tables (views), which in turn, can be operated on again. The standardized language for database queries, SQL (Structured Query Language) is used and this makes it possible to define and manipulate tables and views.

As the database structure is revised it is important that existing programs can be used even after the revision. In a relational database multiple views (combinations of tables or parts of them) of the same data is allowed.

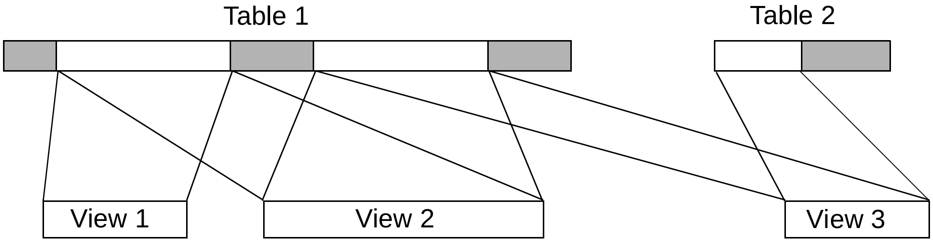


Figure A-4. Three views are created by using data from two tables

In this way, different programs can access different views of the same data which, in turn, promotes data independence. Existing programs can continue to run and access the old views without any modification, while your new applications use the new views.

RDBMS Kernel

The kernel of the RDBMS can be seen as the "control room" of the RDBMS and manages the following tasks:

- Storage and definition of data
- Controlling and limiting data access and concurrence
- Backup and recovery of data
- Interpretations of SQL and PL/SQL commands.

A.2 SQL (Structured Query Language)

SQL is an ANSI-Standardized, English-like database language.

SQL is a non-procedural language. With a non-procedural language, users only have to specify what they want to be done and not how it has to be done. Some example on procedural languages are C, COBOL and Pascal.

SQL can be used in two ways:

- **interactive** (licence included in AdvaInform Basic Functions)
- **embedded**. (Programmer 2000 Pro*C Licence).

Interactive SQL is entered at a terminal and is immediately processed and made visible to the user. SQL*Plus is used to make interactive queries to the database with SQL. (A run-time licence of SQL*Plus is included in the AdvaInform Basic Functions).

With embedded SQL, the different statements are embedded in a program written in another, procedural 3-GL language. The result is not visible to the user and can be further processed.

PL/SQL can be seen as an extension of SQL, allowing entire blocks of statements to be sent to the database. Thus giving optimal access to the database.

An example of an SQL call, assuming that the user has logged on to the Oracle database with the correct password:

```
SELECT * FROM AI WHERE NAME=’AI_1’;
```

Figure A-5. An Example of Interactive SQL

The SQL query described in the figure above, will read the AI object named “`AI_1`” residing in a Controller or locally in the Oracle database by using AdvaInform SQL*Connect. (In AdvaInform Basic Functions this can be done from, for example SQL*Plus).

Figure A-6 shows an example of PL/SQL. This routine displays the value of an object named "`AI_1`" on the terminal display from where the routine is started.

```
DECLARE
     ai1_value FLOAT;
     ai2_value FLOAT;
product FLOAT
BEGIN
     SELECT value INTO ai1_value FROM AI WHERE NAME='AI_1';
     SELECT value INTO ai2_value FROM AI WHERE NAME='AI_2';
     product := ai1_value*ai2_value;
     INSERT INTO product_tab VALUES (product);
END;
```

Figure A-6. An Example of PL/SQL

The Programmer 2000 - Pro*C licence is needed to realize the example above. The difference between these two examples is that SQL embedded in C is a program or procedure, has better performance and does not need to be rewritten every time the query needs to be made. On the other hand, Interactive SQL is more flexible, but all queries need to be rewritten every time they are executed.

Appendix B SQL Statement Reference

B.1 General

SQL language reference for AdvaInform Relational Database gives a description of SQL commands which are allowed to be used towards the Relational Database. Interactive access towards the Relational Database is performed with the Oracle user IMS_USER.

This Appendix is not complete. It only describes a subset of the SQL language. In the Advant OCS price list you can find an Oracle SQL*Plus reference manual. Please refer to that manual for a complete reference.

B.2 CREATE DATABASE LINK

B.2.1 Purpose

Create a link from the local database to a username on a remote database. This database link may reference a remote ORACLE or a remote non-ORACLE database, such as an IBM DB2 database.

B.2.2 Prerequisites

Must have access to a username on a remote database. Only DBAs may create PUBLIC database links. SQL*Net must be installed on both the local and remote databases.

B.2.3 Syntax

```
CREATE [PUBLIC] DATABASE LINK link
    [CONNECT TO user IDENTIFIED BY password]
    USING ‘connect string’
```

B.2.4 Keyword and Parameters

PUBLIC: If the PUBLIC option is omitted, the link is private (available only to its creator). If the PUBLIC option is specified, the link is available to all users except those who have defined a private database link with the same name

`link`	is the name to be given to the database link
`user, password`	are a valid username and password on the specified remote database
`connect_string`	is the database specification of a remote database accessible through SQL*Net.

B.2.5 Usage Notes

Once a database link has been created, you can query a table on the remote database by referring to that link in the FROM clause of a query or subquery, with the same effect as if you were connected directly to the remote database as the user.

If the clause containing the username and password is omitted, the database link will use the current username and password when invoked. For example, if SCOTT connects to his local database and then uses a database link that was not created with a username and password, ORACLE will use SCOTT and SCOTT's password to access the remote database.

Remote tables are referenced in queries by adding @linkname to remote tables within the FROM clause of a SELECT statement.

Queries and subqueries performed through database links are subject to the following restrictions:

- The number of database links that may be used in a single statement is initially limited in most systems to four. The DBA may change the limit by increasing the INIT.ORA parameter OPEN LINKS.
- LONG columns may not be selected through a database link.
- In tree-structured queries (using the CONNECT BY clause):
  - The PRIOR operator may not be used in clauses other than CONNECT BY
  - The START WITH clause may not contain subqueries
  - Neither the CONNECT BY clause nor the START WITH clause may contain the function USERENV('ENTRYID'), user-defined functions, or ROWNUM

B.2.6 Example

To define a database link named BOSTON that refers to user SCOTT with password TIGER on database D:BOSTON-MFG, enter:

```
CREATE DATABASE LINK BOSTON

    CONNECT TO SCOTT IDENTIFIED BY TIGER

    USING ‘D:BOSTON-MFG’;
```

Once this database link has been created, you can query tables owned by SCOTT on D:BOSTON-MFG in the following manner:

```
SELECT * FROM EMP@BOSTON;
```

You can also query tables owned by other users on the same database. This example assumes SCOTT has access to ADAM's DEPT table:

```
SELECT * FROM ADAMS.DEPT@BOSTON;
```

The previous statement would connect to the user SCOTT on the remote database and then query ADAM's DEPT table.

A synonym may be created to hide the fact that SCOTT’s EMP table is on a remote database. The following statement would cause all future references to EMP to access a remote EMP table owned by SCOTT.

```
CREATE SYNONYM EMP FOR SCOTT.EMP@BOSTON;
```

B.2.7 Related Topics

The CREATE SYNONYM and SELECT statements.

B.3 CREATE VIEW

B.3.1 Purpose

Define a view (a logical table based on one or more tables).

B.3.2 Prerequisites

Must own or have SELECT privilege on the base table(s).

B.3.3 Syntax

`CREATE VIEW [user.]view [(alias [,alias]...)]`

AS query

`[WITH CHECK OPTION [CONSTRAINT constraint]]`

B.3.4 Keywords and Parameters

`user`	is the owner of the view to be created.
`view`	is the name of the view.
`query`	identifies columns and rows of the table(s) that the view is based on. The query may be any valid SELECT statement without the ORDER BY and FOR UPDATE clauses. Refer to the syntax description of query in Section B.6, SELECT (Embedded SQL).
`WITH CHECK OPTION`	specifies that inserts and updates performed through the view should not be allowed to create or result in rows that the view cannot select. The WITH CHECK OPTION may be used in a view that is based on another view.
`CONSTRAINT`	is the name assigned to the WITH CHECK OPTION constraint. If no constraint name is supplied, then the constraint is supplied with a system assigned name of the form SYS Cn, where n is an integer that makes the constraint name unique within the system.

B.3.5 Usage Notes

A view is a logical window of one or more tables. Views have the following properties:

- A view stores no data; rather, it displays data from other tables (which may in turn be views).
- A view may be used in a SQL statement anywhere you may use a table (for example, INSERT, UPDATE, DELETE, SELECT) with the restriction that you may only select from views if the view query contains:
  - a join
  - a GROUP BY, CONNECT BY or START WITH clause
  - the DISTINCT clause, pseudo-columns (e.g. ROWNUM) or expressions in the column list

You may update a view that contains pseudo-columns or expressions as long as the UPDATE statement does not refer to any of thc pseudo-columns or expressions of the view.

Views are used to:

- provide an additional level of table security, by restricting access to a predetermined set of rows and /or columns of a base table.
- hide data complexity. For example, a view may be used to act as one table when actually several tables are used to construct the result.
- present the data from another perspective. For example, views provide a means to rename columns without actually affecting the base table's definition.
- provide a level of referential integrity.

B.3.6 Naming Guidelines

The view column names must conform to the following guidelines:

- Column names must follow the standard naming conventions for database objects.
- Column names must use an alias if they derive from expressions in the query.
- Column names must be distinct (tables that comprise a view could have identically named columns).
- Column names will be derived from the query columns unless column aliases are used. Column aliases may also be used in the select list of the query.

  When aliases are used, there must be as many aliases as columns selected in the query. For example, if the query selects four columns, then there must be four aliases.
  The first query column will be assigned the first alias, the second query column will be assigned the second alias, and so on.

When you access a view, ORACLE performs the following steps:

1. Retrieves the view definition as specified in the CREATE VIEW statement and stored in the data dictionary.
2. Checks your access privileges.
3. Uses the view query to retrieve the data from the base table(s). A base table is any table listed in the view query.

NOTE

If a view is defined with the "AS SELECT * FROM tablename" construct and the table is subsequently altered to add new columns, then the view will no longer work. This may be corrected by dropping and re-creating the view.

B.3.7 AUDITING VIEWS

The default audit options for a view are the union of all the audit options for the tables that comprise the view.

B.3.8 Examples

To create a view named DEPT20 of table EMP that shows the employees in Department 20 and their annual salary, enter:

```
CREATE VIEW DEPT20

AS SELECT ENAME, SAL*12 ANNUAL SALARY

FROM EMP

WHERE DEPTNO = 20;
```

NOTE

A view column alias was not required for the SAL*12 expression because a query column alias (ANNUAL SALARY) was used.

To create a view named ROSTER of all employees in EMP that shows only their IDs, names, titles, and department number (and includes the WITH CHECK OPTION constraint named WCO), enter:

```
CREATE VIEW ROSTER (ID NUMBER, PERSON, TITLE)

AS SELECT EMPNO, ENAME, JOB, DEPTNO

FROM EMP

WHERE DEPTNO IN (SELECT DISTINCT DEPTNO FROM DEPT)

WITH CHECK OPTION CONSTRAINT WCO;
```

Because of the WITH CHECK OPTION, ORACLE would prevent you from inserting a new row into the view ROSTER if the new employee's department number did not exist in the DEPT table. Note that the row would actually be inserted into the EMP table only after passing the WITH CHECK OPTION test.

B.3.9 Related Topics

CREATE TABLE, CREATE SYNONYM, DROP VIEW, and RENAME statements.

B.4 DESCRIBE (Dynamic Embedded SQL: Method 4)

B.4.1 Purpose

Allocate and initialize a descriptor to hold the host variable descriptions from a SQL statement previously prepared in a PREPARE statement.

B.4.2 Prerequisites

You must PREPARE the SQL statement that contains the host variables to be described.

B.4.3 Syntax

For bind variables:

```
EXEC SQL DESCRIBE BIND VARIABLES FOR statement name
    INTO descriptor name
```

For SELECT-list entries:

```
EXEC SQL DESCRIBE [SELECT LIST FOR] statement name
    INTO descriptor name
```

B.4.4 Keywords and Parameters

`BIND VARIABLES`	is required to specify that the statement should describe input variables.
`SELECT LIST`	specifies that a output variable list should be described. The explicit use of this clause is optional.
`statement_name`	indicates a previously prepared SQL statement may contain valid references to host variable arrays.
`descriptor_name`	indicates the name by which the descriptor is referenced.

B.4.5 Usage Notes

The descriptor must be previously declared in the program. Output and input variables should not be included in the same descriptor.

The DESCRIBE is generally executed after the PREPARE statement.

B.4.6 Example

For a language specific example, refer to the Supplement to the ORACLE Precompilers Guide for your specific programming language.

B.4.7 Related Topic

PREPARE statement; Supplement to the ORACLE Precompilers Guide.

B.5 SELECT

B.5.1 Purpose

Display rows and columns from one or more tables. May be used as a statement, or (with certain restrictions on clauses) as a query or subquery in another statement.

B.5.2 Prerequisites

Must own the table, or have SELECT privilege on the table(s), or have DBA privilege.

B.5.3 Syntax

```
SELECT [ALL|DISTINCT] {* I table.* I expr [c alias]}
    FROM [user.]table [t alias] [,[user.]table [t_ alias]]...
    [WHERE condition]
    [CONNECT BY condition [START WITH condition]]
    [GROUP BY expr[, expr]... [HAVING condition]]
    [{UNION|INTERSECT|MINUS} SELECT... j
    [ORDER BY {expr|position} [ASC|DESC]
    [, [expr|position} [ASC|DESC]]]...
    [FOR UPDATE OF column [, column]... [NOWAIT]]
```

B.5.4 Keywords and Parameters

`ALL`	is the default and specifies that all rows that satisfy the SELECT statement should be displayed.
`DISTINCT`	specifies that duplicate rows should be removed before the rows are returned. A row is considered a duplicate of another if every value for each column of the SELECT clause matches that of another row(s).
`*`	specifies that all columns from all tables listed in the FROM clause to be displayed.
`table.*`	specifies that all columns from the named table to be displayed.
`expr`	Refer to the syntax description of expr in Chapter 4, Runtime Operation.

`c_alias`	provides a different name for the column expression and causes the alias to be used in the column heading. Does not affect the actual name of the column. Column aliases cannot be referenced elsewhere in a query.
`[user.]table`	specifies which table(s) or view(s) to display, optionally prefaced by the owner name of the table. If not specified, the owner name defaults to the current user.
`t_alias`	provides a different name for the table for the purpose of evaluating the query and is most often used in a correlated query. Other references to the table throughout the query must refer to the alias name.
`condition`	Refer to the syntax description of condition in Chapter 4, Runtime Operation.
`positions`	identifies a column based on its relative position in the SELECT column list rather than by its name.
`ASC|DESC`	specifies the column should be sorted ascending or descending based on the operating system collating sequence.
`column`	is a column belonging to a table listed in the FROM clause.
`NOWAIT`	specifies that ORACLE should return control to the user, rather than waiting, if it attempts to lock a row that is locked by another user.

Many of the SELECT clauses are described in Section B.6, SELECT (Embedded SQL).

B.5.5 Usage Notes

Clauses must be used in the order shown above except that:

- CONNECTED BY, START WITH, GROUP BY, and HAVING may be specified in any order.
- ORDER BY and FOR UPDATE OF may be interchanged.

Each expr becomes the name of one column in the display, and each table.* becomes a set of columns, one for each column in table in the order they were defined when the table was created. The width and datatype of each expression is determined by the elements of the expression.

The asterisk character, *, represents all columns in a table. If multiple tables are specified in the FROM clause, then * represents all columns from all tables. The asterisk may be prefaced by table., which limits the columns to those belonging to the named table.

Each alias, if specified, is used to label the preceding expr in the table so that the column is displayed with a new heading. An alias has no effect on the actual name of the column, and cannot be referenced in the remainder of the query.

FROM table specifies the table or view to be selected from. More than one table implies a join. A table alias value, if specified, must be used as an alias for the table throughout the entire SELECT statement.

WHERE specifies a condition to be used to select rows and /or join tables. This condition may often contain subqueries a nested SELECT statement.

B.5.6 CONNECT BY

CONNECT BY is used for querying hierarchical relationships. CONNECT BY specifies that rows are to be retrieved in a hierarchical order, and defines the relationship to be used to connect table rows into a hierarchy. The PRIOR clause must be used before one side of the condition. The PRIOR side represents the parent in each parent-child relationship between nodes; the other side represents the child. For example, CONNECT BY PRIOR EMPNO = MGR means that the EMPNO field in a parent node equals the MGR field in each of its child node. The number of CONNECT BY levels is limited by available user memory.

You may not use CONNECT BY in conjunction with:

- Subqueries
- Joins.

B.5.7 START WITH

START WITH identifies the row(s) to be used as the root(s) of the tree by specifying a condition that they must satisfy. The omission of START WITH is identical to starting with all rows that satisfy the WHERE clause of the SELECT statement.

Subqueries are permitted in a START WITH clause.

When a SELECT statement contains the CONNECT BY clause, it may use the pseudo-column LEVEL. LEVEL returns a number value of 1 for a root node, 2 for a child node of a root node, 3 for a grandchild, and so on.

Refer also to the START WITH clause in Section B.6, SELECT (Embedded SQL).

B.5.8 GROUP BY, HAVING

The GROUP BY and HAVING clauses are used to display summary information about groups of rows that have the same values in one or more fields. Refer to the GROUP BY clause in Section B.6, SELECT (Embedded SQL).

B.5.9 Set Operators

The UNION, INTERSECT, and MINUS operators combine the results of two SELECT statements into a single result. UNION returns those results returned by either component query and eliminates duplicate results; INTERSECT returns only those rows returned by both queries; and MINUS returns those rows returned by the first SELECT after removing those rows that also are returned by the second SELECT. The number and types of the columns selected by each component query must be the same (but the lengths need not be the same).

If more than two SELECTs are combined in this way, adjacent pairs of SELECTs are evaluated from the top down. (Parentheses may be used to specify evaluation in a different order.)
All three operators imply DISTINCT in all SELECTs.

B.5.10 ORDER BY

The ORDER BY clause specifies the order in which rows of the table are returned. Refer to the ORDER BY clause in Section B.6, SELECT (Embedded SQL).

B.5.11 FOR UPDATE OF

The FOR UPDATE clause locks the selected rows of the tables. The SELECT... FOR UPDATE statement is normally followed by one or more UPDATE...WHERE statements. Once you have locked a row, other users cannot lock or update it until you free it with a COMMIT or ROLLBACK statement. (Other users only see changes after you have commit them.)

FOR UPDATE signals that you intend to update the rows returned by the SELECT, but does not require you to update the rows. The term "update" includes the UPDATE. INSERT and DELETE statements

SELECT... FOR UPDATE cannot be used with:

- DISTINCT or GROUP BY
- UNION, INTERSECT, or MINUS operators
- Set functions like COUNT and MAX

NOWAIT makes SELECT terminate, rather than wait, if a row selected by FOR UPDATE OF cannot be locked because someone else is using it.

B.5.12 Sequences

A SELECT statement may also refer to the sequence pseudo-columns NEXTVAL and CURRVAL. For a complete definition of sequences and their use, refer to the CREATE SEQUENCE statement earlier in this chapter.

B.5.13 Subquery

Refer to the syntax description of query in Section B.6, SELECT (Embedded SQL).

B.5.14 Examples

To display ENAME, SAL, JOB, and DEPTNO from all rows of the table EMP in which the department number is 30 or less, enter:

```
SELECT ENAME, SAL, JOB, DEPTNO
    FROM EMP
    WHERE DEPTNO=30
```

B.5.15 Related Topics

The FROM, WHERE, GROUP BY, HAVING, ORDER BY clauses; the syntax descriptions of simple join, outer join, and query in, Section B.6, SELECT (Embedded SQL).

B.6 SELECT (Embedded SQL)

B.6.1 Purpose

Retrieve rows from the specified table. The values returned are assigned to the output host variables.

B.6.2 Prerequisites

Must own the table, or have SELECT privilege for the table, or have DBA privilege.

B.6.3 Syntax

```
EXEC SQL [AT dbname] [FOR :host integer]
SELECT select list
    INTO :host variable [,:host variable]...
    FROM table list
    [WHERE condition]
    [CONNECT BY condition [START WITH condition]]
    [GROUP BY expr[, expr]... [HAVING condition]]
    [{UNION|INTERSECT|MINUS} SELECT...]
    [ORDER BY {expr|position} [ASC|DESC]
    [, ~expr|position} [ASC|DESC]]].
    [FOR UPDATE OF column [, column]... [NOWAIT]]
```

B.6.4 Keywords and Parameters

WHERE

`AT dbname`	is an optional clause that refers to a previous CONNECT statement. This database name must have been previously declared in a DECLARE DATABASE statement.
`FOR:`	Iimits the maximum number of rows fetched. Refer to the FOR clause `host_integer` in Section B.7, SELECT STATEMENT CLAUSES.
`select_list`	identical to the normal SELECT statement except that a host variable may be used wherever a constant may be specified.
`INTO`	introduces the list of output host variables. The host variable is declared as a normal procedural language variable and is either predefined or user-defined for use in a SQL statement. If any host variable in the INTO list is an array, all host variables must be arrays, but they are not required to be arrays of the same size.
`FROM`	identical to the non-embedded SELECT statement.

NOTE

You may not use host variables for table names.

`WHERE`	may contain references to host variables, but may not reference arrays.
`condition`	is a logical expression containing references to host variables or host arrays.

All other keywords and parameters are identical to the non-embedded SQL SELECT statement.

B.6.5 Usage Notes

The embedded SELECT statement is most often used to return one row at a time, but when using arrays as host variable targets, may retrieve multiple rows per FETCH.

If no row is returned, the SELECT statement assigns the value +100 (+141)3 in Version 5 compatibility mode) to sqlcode and does not assign values to the host variables.
The WHENEVER statement can be used to specify the conditional flow of control when the NOT FOUND (+100) condition (no rows returned) occurs.

The SELECT statement can be used in a DECLARE CURSOR or PREPARE statement in conjunction with a later FETCH statement. In these cases, it may include a FOR UPDATE OF clause:

```
EXEC SQL DECLARE cl CURSOR
FOR SELECT column [,column,...]
    FROM table|view [, table|view] [,...]
    WHERE search condition
    FOR UPDATE OF column [, column] [,...]
```

The column names specified can then be referenced in a CURRENT OF clause of a UPDATE statement which would follow the FETCH statement. For details, refer to the DECLARE CURSOR statement earlier in this chapter

B.6.6 Examples

The following example illustrates the use of embedded SELECT:

```
EXEC SQL SELECT ENAME, SAL + 100, JOB
    INTO :ENAME,:SAL,:JOB
    FROM EMP
    WHERE EMPNO =:EMPNO;
```

The following example illustrates the use of a SELECT statement:

```
EXEC SQL INSERT INTO EMP (ENAME,DEPTNO)
    SELECT * FROM NEWHIRES
```

B.6.7 Related Topics

PREPARE, DECLARE CURSOR, FETCH, and EXECUTE statements.

B.7 SELECT STATEMENT CLAUSES

In this section, each clause of the SELECT statement is described in detail. Additional uses of the SELECT statement are also covered.

Topics include:

- FROM clause
- WHERE clause
- GROUP BY and HAVING clauses
- ORDER BY clause
- CONNECT BY clause
- Simple joins
- Outer joins
- Query
- Correlated queries
- Distributed queries.

B.7.1 FROM clause

B.7.1.1 Purpose

Specifies a table or tables for a SELECT or DELETE statement.

B.7.1.2 Prerequisites

None.

B.7.1.3 Syntax

In DELETE statements:

```
DELETE [FROM] [user.] table [@databaselink] [alias]
[WHERE...]
...
```

In DELETE statements:

```
SELECT select list
FROM [user.] table [alias] [@databaselink] [,...]
WHERE...
...
```

B.7.1.4 Keywords and parameters

user	is the owner of the table or view.
`table`	is the name of the table or view.
`alias`	a temporary name assigned to a table or column an alias allows the same table to be referenced in a correlated query. The alias also allows you to abbreviate a table name within the query.
`databaselink`	is a valid database link that references a remote database. Refer to the CREATE DATABASE LINK statement.

B.7.1.5 Usage Notes

FROM is a required clause in the SELECT statement and is optional in the DELETE statement. It identifies the table from which rows are deleted, or the table(s) from which rows are selected.

The REVOKE statement has a FROM clause which serves other functions. This description does not apply to that statement.

If the table is owned by another user, the table name must be qualified with the owner's username. The username and table name are separated by a period.

B.7.1.6 Examples

To delete rows with the value 50 in the DEPTNO column from a table

named EMP, enter:

```
DELETE FROM EMP
    WHERE DEPTNO = 50;
```

To select rows with the value 40 in the DEPTNO column from a table named EMP, enter:

```
SELECT * FROM EMP
    WHERE DEPTNO = 40;
```

To select rows with the value 40 in the DEPTNO column from two tables named EMP and EMP2, enter:

```
SELECT *
    FROM EMP, EMP2
    WHERE EMP.EMPNO = EMP2.EMPNO
    AND DEPTNO = 40;
```

If EMP2 is owned by another user whose user ID is BEN, enter:

```
SELECT *
    FROM EMP, BEN EMP2
    WHERE EMP.EMPNO = EMP2.EMPNO
    AND DEPTNO = 40;
```

B.7.1.7 Related Topics

SELECT and DELETE statements in this appendix.

B.7.2 WHERE clause

B.7.2.1 Purpose

Restrict which rows are returned from the table in a SELECT statement, or acted upon in a DELETE or UPDATE statement.

B.7.2.2 Prerequisites

None.

B.7.2.3 Syntax

DELETE...

WHERE condition

SELECT...

WHERE condition

UPDATE...

WHERE condition

B.7.2.4 Keywords and Parameters

None

`condition`	(Refer to the syntax description for condition in Chapter 4 "Operators and Functions" of this Manual.)

B.7.2.5 Usage Notes

The purpose of a WHERE clause is to specify a condition. The statement containing the WHERE clause will act only on rows that satisfy the condition.

The condition is defined by a logical expression. For example,

```
DEPTNO=30
```

is a logical expression that is true if the value of DEPTNO is 30, and otherwise is false.

AND and OR may be used to form compound logical expressions. For example, the expression:

```
DEPTNO=30 AND JOB=′ SALESMAN’
```

is true only if both of the component expressions are true.

NOT may be used to reverse the sense of an expression. For example,

```
NOT (DEPTNO=30 AND JOB=′ SALESMAN′
```

is true if:

```
DEPTNO=30 AND JOB=′ SALESMAN′
```

is false.

B.7.2.6 Examples

This statement selects all rows in which the value of DEPTNO is not equal to 30:

```
SELECT *
    FROM EMP
    WHERE DEPTNO!=30;
```

This statement updates all rows in which the value of COMM is null:

```
UPDATE FROM DEPT A
    SET SAL = (1.1*SAL)
    WHERE COMM IS NULL;
```

This statement uses a correlated query to delete all departments that have no employees:

```
DELETE FROM DEPT A
    WHERE NOT EXISTS
    (SELECT ‘TRUE’
    FROM EMP B
    WHERE A. DEPTNO = B. DEPTNO;
```

For every row in the DEPT table, ORACLE performs the following steps:

1. Determine the row's DEPTNO.
2. Use that row's DEPTNO to determine if that DEPTNO exists in the EMP table.
3. If the DEPTNO does not exist in the EMP table, delete the row. If the DEPTNO exists for any of the rows of the EMP table, do not delete the row.

Note that this DELETE uses a correlated query because the query refers to a table in the parent query. The 'TRUE' constant is a place holder and can be replaced by any constant or column from the EMP table. The NOT EXISTS operator specifies that the row in question will be deleted if no rows are returned by the query.

B.7.2.7 Using the TO_DATE Function in a Condition

The TO_DATE function has the following syntax:

```
TO_DATE(char[,fmt])
```

The purpose is to convert *char* to a date value. The current month, year, and so on, are used to provide components of the date not explicitly determined from the arguments. *fmt* is a char value in date format specifying the format of *char*. If *fmt* is omitted, *char* must have the default format, "DD-MON-YY". IF *fmt* is J, for Julian, then *char* should actually be a number.

In Section 4.4.2, How to Execute Queries in the Database Access Window, provides an example on how to use the TO_DATE format in a SELECT statement to access Numeric Log data.

Date formats

Date formats are used in the TO_DATE function (as shown above) to supply values to Oracle in a nonstandard date format. As mentioned above the standard date format is DD-MON-YY. Table B-1 lists the elements of a date format model.

Table B-1. Date formats in the TO_DATE function

Element	Meaning
SCC or CC	Century;"S" prefixes "BC" date with "-"
YYYY or SYYY	4-digit year; "S" prefixes "BC" date with "-"
YYY or YY or Y	Last 3,2, or 1 digit(s) of year
Y,YYY	Year with comma in this position
SYEAR or YEAR	Year, spelled out; "S" prefixes "BC" date with "-"
BC or AD	BC/AD indicator
B.C. or A.D.	BC/AD indicator with periods
Q	Quarter of year (1,2,3,4; JAN-MAR=1)
MM	Month (01-12; JAN=01)

Table B-1. Date formats in the TO_DATE function (Continued)

Element	Meaning
MONTH	name of month, padded with blanks to length of 9 characters
MON	Name of month, 3-letter abbreviation
WW	Week of year (1-52) where week 1 starts on the first day of the year and continues to the seventh day of the year
W	Week of month (1-5) where week 1 starts on the first of the month and continues to the seventh of the month
DDD	Day of year (1-366)
DD	Day of month (1-31)
D	Day of week (1-7)
DAY	Name of day, padded with blanks to length of 9 characters
DY	name of day, 3-letter abbreviation
J	Julian day; the number of days since january 1, 4712 BC
AM or PM	Meridian indicator
AM or PM	Meridian indicator with periods
HH or HH12	Hour of day (1-12)
HH24	Hour of day (1-23)
MI	Minute (0-59)
SS	Second (0-59)
SSSSS	Seconds past midnight (0-86399)
/.,.	Punctuation is reproduced in the result
...	Quoted string is reproduced in the result

Prefix *FM* - "Fill Mode":

This prefix suppresses blank padding for all subsequent character elements (such as MONTH) and suppresses leading zeroes for all subsequent number elements (such as MI) in a date format model. FM is a toggle. If FM appears twice in a date format model, blank padding and leading zeroes are suppressed for elements after the first FM and re-enable for elements after the second.

Capitalization in a spelled-out word or abbreviation follows capitalization in the corresponding format element. For example, "DAY" in a format model produces capitalized words like "MONDAY"; "Day" produces "Monday"; and "day" produces "monday".

A date format model may also include punctuation characters like hyphens, slashes, and commas, and character constants enclosed in quotation marks (not apostrophes). Such punctuation characters and character constants will be reproduced in the output.

For example, the format model:

```
‘DDTH “of” fmMonth, YYYY’
```

produces a date like:

15TH of March, 1989

Note that the fill mode (*fm*) format mask was also used. If omitted, the month would be blank padded to 9 characters, as in:

15TH of March, 1989

To include an apostrophe in a date format element, use two apostrophes in a row. For example:

```
SELECT TO_CHAR (sysdate, ‘fmDay”’’s Special”’)
   FROM DUAL
```

would yield:

Tuesday’s Special

The syntax description of query later in this chapter; the section “Operators” in Chapter 4 “Operators and Functions”; DELETE, SELECT, and UPDATE statements in “SQL Statement Reference” of this Manual.

B.7.3 WHERE clause (Embedded SQL)

B.7.3.1 Purpose

Restrict which rows are returned from the table in SELECT, DELETE, or UPDATE statements.

B.7.3.2 Prerequisites

None.

B.7.3.3 Syntax

The syntax for an embedded WHERE clause is as follows:

```
SELECT...

   FROM...

   WHERE [condition I CURRENT OF cursor name]
```

B.7.3.4 Keywords and condition Parameters

`condition`	is a logical expression that may refer to host variables or host variable arrays (the WHERE clause of the SELECT may not contain host variable arrays). The search condition may contain a query. Refer to the syntax description of query later in this appendix.
`CURRENT OF`	
`cursor_name`	can only be used with DELETE or UPDATE; it refers to the row last processed by the FETCH statement associated with the named cursor.

B.7.3.5 Usage Notes

If a WHERE clause is omitted, all rows of the table or view are processed. Host variable references may be used in the statement anywhere constants are allowed.

If the WHERE clause is omitted in a DELETE or UPDATE statement, the 5th component of sqlwarn (SQLCA) is set to 'W' as a warning and the EXEC SQL WHENEVER SQLWARNING action is taken.

If all host variables used in the WHERE clause search condition are arrays, the result is the same as if the statement had been executed once for each set of array components.

If any of the host variables used in the WHERE clause are arrays, then all of the host variables in the WHERE clause must be arrays.

The arrays may be of different sizes, in which case the smallest size (or the value specified in an optional FOR clause) is used to specify the number of times the SQL statement containing the WHERE clause is executed.

The WHERE clause of the SELECT statement may not contain host variable arrays.

B.7.3.6 CURRENT OF Clause

If the CURRENT OF clause is used in an UPDATE statement, then the cursor must have been previously declared using the FOR UPDATE clause in the SELECT statement. The cursor must be open and positioned on a row. If no FETCH has been made or if the cursor is not open, the CURRENT OF clause results in an error and returns zero rows.

B.7.3.7 Example

To update the current row of the EMP table and set salary to 2000, enter:

```
UPDATE EMP
    SET SAL = 2000
    WHERE CURRENT OF CURSOR EMP CURSOR;
```

B.7.3.8 Related Topics

FOR clause and DECLARE CURSOR, FETCH, EXECUTE, and SELECT statements in Chapter 5 “SQL Statement Reference” of this Manual.

B.8 UPDATE

The tables available for UPDATE for the IMS_USER are DB_AI, DB_AO, DB_DI, DB_DO, DB_DAT and DB_TEXT. No other tables in the IMS Relational Database are allowed to be updated.

B.8.1 Purpose

Change the data in the specified table.

B.8.2 Prerequisites

You must own the table, or have UPDATE privilege on the table, or have the DBA privilege.

B.8.3 Syntax

```
UPDATE [user. ] table [alias]
       SET column = expr [,column. = expr]...
       [WHERE condition]
```

or

```
UPDATE [user.]table [alias]
       SET (column [,column] ...) = (query)
       [, (column [,column] ...) = (query) ]
       [WHERE condition]
```

B.8.4 Keywords and Parameters

`user`	is name of the owner of the table.
`table`	is the name of an existing table that you can access.
`alias`	is used to rename the reference in the other clauses of the statement.
`column`	is a column within the table. No parentheses are required if the column list contains only one column.
`expr`	A text string, number, or date value or expression.
`query`	is a SELECT statement without the ORDER BY or FOR UPDATE clause, often correlated with the table being updated.
`condition`	is a valid condition that evaluates to TRUE or FALSE.

B.8.5 Usage Notes

The SET clause determines which columns are updated and what new values are stored in them.

The WHERE clause determines the rows in which values are updated. If no WHERE clause is specified, all rows are updated. For each row that satisfies the WHERE clause, the columns to the left of the equals (=) operator in the SET clause are set to the values of the corresponding expressions on the right. The expressions are evaluated as the row is being updated.

Query

If the SET clause contains a subquery, it must return exactly one row for each row being updated. Each value in the query result is assigned respectively to the columns in the parenthesized list. If the query returns no rows, then the value (for example, SET value = query) will be set to NULL. Subqueries may select from the table being updated.

The SET clause may intermix assignments of expressions and subqueries.

The UPDATE statement automatically obtains exclusive locks on the rows or table being updated. These locks are released after a COMMlT or ROLLBACK statement is issued.

Correlated Update

If a subquery refers to columns from the table being updated, ORACLE will evaluate the subquery once for each row, rather than once for the entire update. This type of updated is call a correlated update. The reference to columns from the table being updated is usually accomplished by means of a table alias.

Potentially, each row evaluated by the UPDATE statement could be updated with a different value as determined by the correlated subquery. Normal UPDATE statements will update each row with the same value.

Rollback Segments

Rollback segments are used by the relational database to save transactions until they are committed. Transactions should be committed as soon as possible to avoid problems with full rollback segments.

For example, instead of dropping a complete tablespace with contents, should all tables be dropped and committed separately or at least in moderate groups to avoid rollback segment problems.

B.8.6 Examples

To set null commissions for all employees for whom the JOB column contains TRAINEE, enter:

```
UPDATE EMP
SET COMM = NULL
WHERE JOB = ‘TRAINEE’;
```

To promote JONES to manager of Department 20 with a $1,000 raise, (and assuming that you know there is only one JONES) enter:

```
UPDATE EMP
SET JOB = ‘MANAGER’, SAL=SAL+1000, DEPTN0=20
WHERE ENAME = ‘JONES’;
```

The following UPDATE shows:

- how the two forms of SET may be mixed in a single statement
- the use of a correlated query
- the use of a WHERE clause to limit which rows are updated.

```
UPDATE EMP A
SET DEPTNO =
(SELECT DEPTNO
FROM DEPT
WHERE LOC = ‘BOSTON’,
(SAL, COMM) =
(SELECT l.l*AVG(SAL3, l.SAVG(COMM)
FROM EMP B
WHERE A.DEPTNO = B.DEPTNO)
WHERE DEPTNO IN
(SELECT DEPTNO
FROM DEPT
WHERE LOC = ‘DALLAS’
OR LOC = ‘DETROIT’);
```

The preceding update:

- Updates only those employees who work in Dallas or Detroit (WHERE clause)
- Sets DEPTNO for these employees to the DEPTNO of Boston (subquery)
- Sets each employee’s salary and commission to 1.5 times the average salary of their current department (correlated subquery).

B.8.7 Related Topics

DELETE and INSERT statements; WHERE clause.

B.9 INSERT

The only tables available for UPDATE for the IMS_USER are DB_AI, DB_AO, DB_DI, DB_DO, DB_DAT and DB_TEXT. No other tables in the AdvaInform Relational Database are allowed for insert of new rows.

B.9.1 Purpose

Add new rows to the specified table or to the table underlying the view.

B.9.2 Prerequisites

You must own the table, or have INSERT privilege for the table, or have DBA privilege.

B.9.3 Syntax

```
INSERT INTO [user. ] table [ (column [, column I . . . ) ]
       { VALUES (value [,value]...) I query )
```

B.9.4 Keywords and Parameters

`user`	is the username for the owner of the table or view.
`tabl`	is a name of an existing table, view, or synonym. The default user is the user entering the INSERT.
`column`	is a column within the table or view.
`value`	is a value that corresponds by position to the list of columns. A value can be any expression. Unless value is NULL, it must match its column's datatype.
`query`	is a SELECT statement that returns as many values as are listed in the column list. It cannot have an ORDER FOR nor FOR UPDATE clause.

B.9.5 Usage Notes

The INSERT statement used with the VALUES keyword always adds just one row. The fields in the specified columns (or all columns) are assigned the specified values. Columns not specified in the column list are assigned NULL and must not have been declared NOT NULL.

The INSERT statement used with query instead of VALUES clause may add multiple rows (all those returned by the query). After the query has been processed, the columns in its result are matched by position to the columns specified after INSERT (or to all columns, if none are specified). The query may reference the target table.

All columns that are defined NOT NULL must be supplied a value. Currently, all columns default to null.

If a parenthesized list of columns is given, values must be provided for those columns; if not, values must be provided for all columns. In the case in which a parenthesized list of columns is not specified, the values provided are matched to the table's column based on the internal position of the columns within the table. The position of each column within the table can be determined by selecting from the USER TAB COLUMNS Data Dictionary view.

No rows will be inserted if the subquery of an INSERT... SELECT returns no rows.

B.9.6 Examples

To insert a new row into the table DEPT, enter:

```
INSERT INTO DEPT VALUES (50, ‘PRODUCTION’, ‘SAN
FRANCISCO’);
```

To insert a row with six columns into the table EMP, including a NULL value and a number in scientific notation, enter:

```
INSERT INTO EMP (EMPNO, ENAME, JOB, SAL, COMM, DEPTNO)
  VALUES (7890, ‘JINKS’, ‘CLERK’, 1.2E3, NULL, 40);
```

To copy managers and presidents or employees whose commission exceeds 25% of their salary into the table BONUS, enter:

```
INSERT INTO BONUS
  SELECT ENAME, JOB, SAL, COMM
    FROM EMP
    WHERE COMM < 25 * SAL
      OR JOB IN (‘PRESIDENT’, ‘MANAGER’);
```

B.9.7 Related Topics

DELETE and UPDATE statements.

B.10 DELETE

This command deletes rows from the specified *table*.

B.10.1 Syntax

```
DELETE FROM tabl [ WHERE condition ]
```

If `WHERE` is specified, all rows where condition is true are deleted. If it is omitted, all rows are deleted.

B.10.2 Example

To delete all rows from a table named `TEMP_ASSIGN`, enter:

```
DELETE FROM TEMP_ASSIGN;
```

To remove from EMP all sales staff who made less than $100 commission last month, enter:

```
DELETE FROM EMP
    WHERE JOB='SALESMAN'
    AND COMM < 100;
```

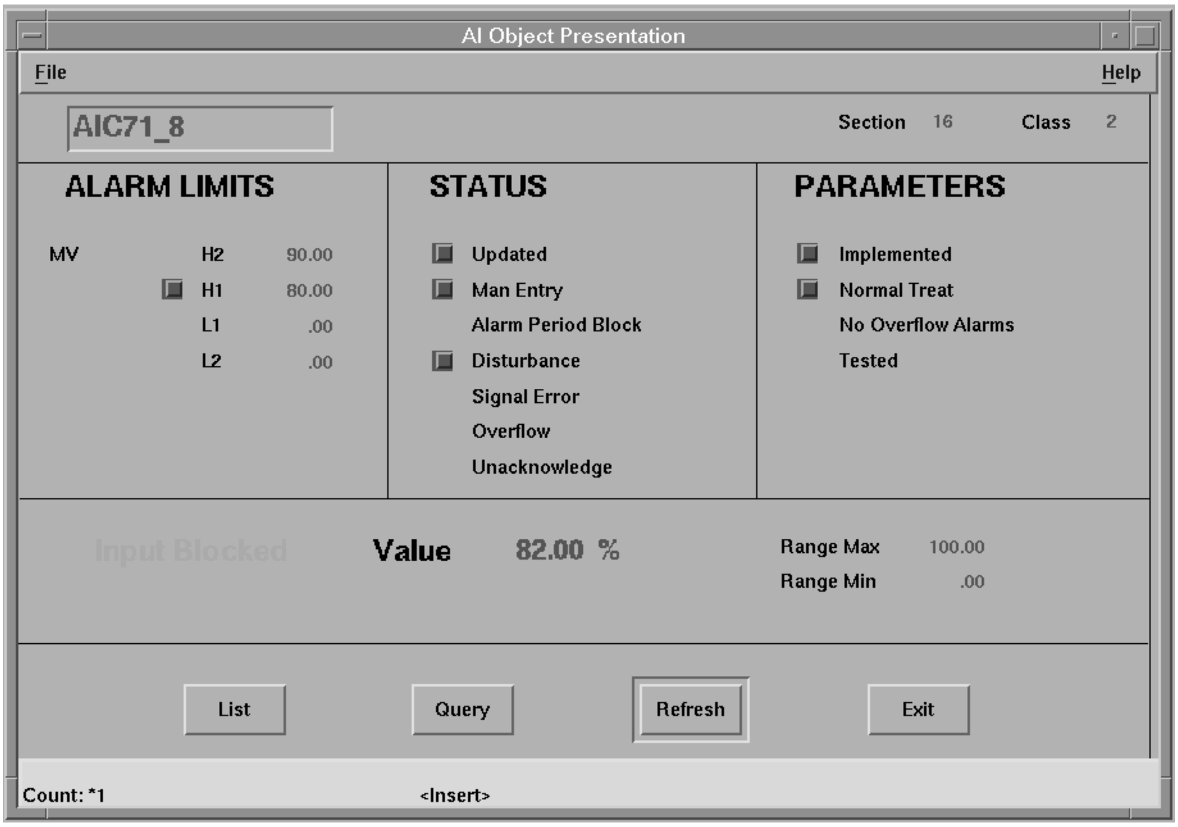
B.10.3 Related Topics

The `FROM` clause; the `WHERE` clause.

Appendix C Object Presentations

C.1 AI Object Presentation

The AI Object Presentation presents information for selected Analog Input objects, Figure C-1



AIC71_8

Section 16 Class 2

MV H2 90.00
■ H1 80.00
L1 .00
L2 .00

■ Updated
■ Man Entry
Alarm Period Block
■ Disturbance
Signal Error
Overflow
Unacknowledge

■ Implemented
■ Normal Treat
No Overflow Alarms
Tested

Value 82.00 %

Range Max 100.00
Range Min .00

Count: *1 <Insert>

Figure C-1. AI Object Presentation

The operation of this presentation is described in the following table.

Description	Presentation	Condition	Remarks
HEADER			*Header*
Name field		NAME	*Object Name*
	Yellow	BLOCKED = 1	*Input blocked*
	Red	DISTURB = 1 or ERR = 1	*Alarm*

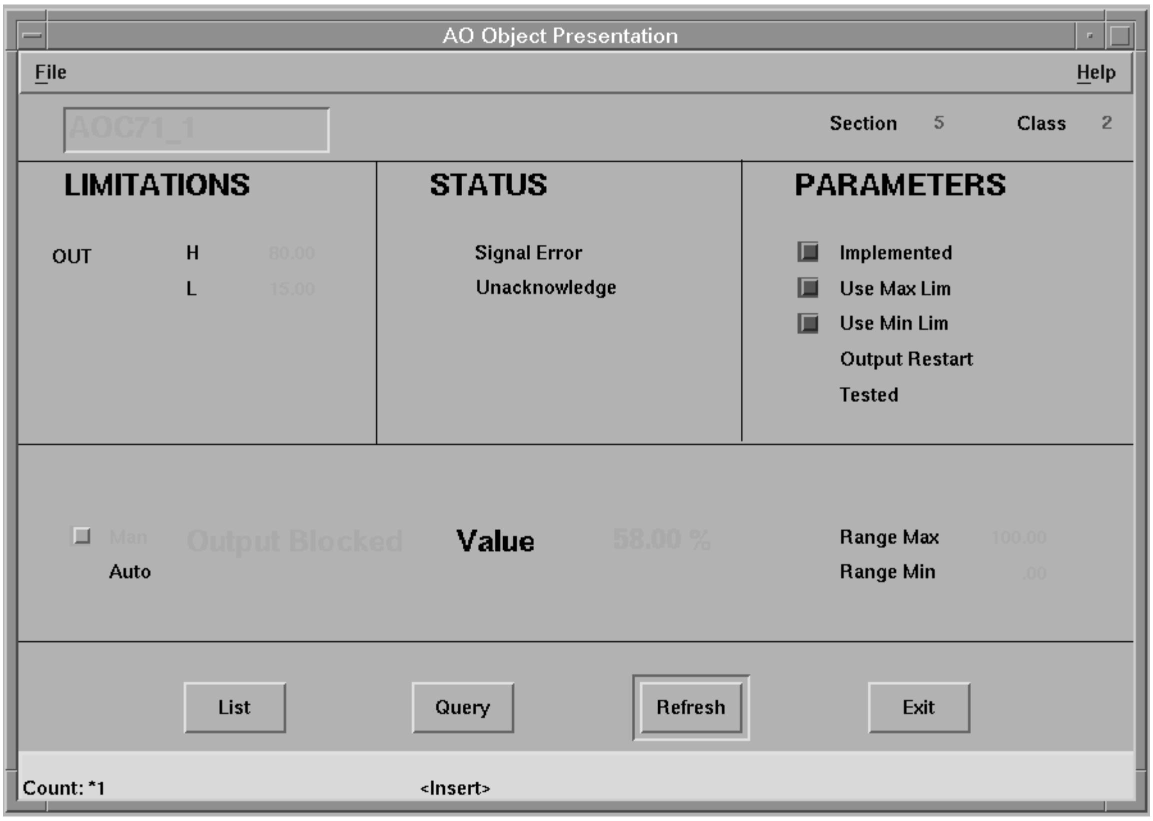
Description	Presentation	Condition	Remarks
	Green	-	*Normal*
Description		DESCR	*Object Description*
	Yellow	BLOCKED = 1	*Input blocked*
	Red	DISTURB = 1 or ERR = 1	*Alarm*
	Green	-	*Normal*
Section text	**Section** Black		*Text in front of the value*
Section value	Green	PROC_SEC	*Process section*
Class text	**Class** Black		*Text in front of the value*
Class value	Green	CLASS	*Object class*
ALARM LIMITS			*Alarm limits*
MV text	**MV** Black		
Printout Blocked	**P** Yellow	PR_BLK = 1	*Printout blocked*
Alarm Blocked	**B** Yellow	AL_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_P_BLK = 1	*Alarm blocked by PC-program*
Alarm Indication	■ Red	VALUE>H2 = 1	*Alarm High limit 2*
H2 text	**H2** Black		*Text in front of the value*
H2 value	Green	HI_LIM2	*High limit 2 value*
	-	EN_H2 = 0	*Invisible*
Printout Blocked	**P** Yellow	PR_BLK = 1	*Printout blocked*
Alarm Blocked	**B** Yellow	AL_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_P_BLK = 1	*Alarm blocked by PC-program*
Alarm Indication	■ Red	VALUE>H1 = 1	*Warning High limit 1*
H1 text	**H1** Black		*Text in front of the value*
H1 value	Green	HI_LIM1	*Warning High limit 1 value*
	-	EN_H1 = 0	*Invisible*
Printout Blocked	**P** Yellow	PR_BLK = 1	*Printout blocked*
Alarm Blocked	**B** Yellow	AL_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_P_BLK = 1	*Alarm blocked by PC-program*
Alarm Indication	■ Red	VALUE<L1 = 1	*Warning Low limit 1*
L1 text	**L1** Black		*Text in front of the value*
L1 value	Green	LO_LIM1	*Warning Low limit 1 value*
	-	EN_L1 = 0	*Invisible*
Printout Blocked	**P** Yellow	PR_BLK = 1	*Printout blocked*

Description	**Presentation**	**Condition**	**Remarks**
Alarm Blocked	**B** Yellow	AL_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_P_BLK = 1	*Alarm blocked by PC-program*
Alarm Indication	■ Red	VALUE<L2 = 1	*Alarm Low limit 2*
L2 text	**L2** Black		*Text in front of the value*
L2 value	Green	LO_LIM2	*Alarm Low limit2 value*
	-	EN_L2 = 0	*Invisible*
STATUS			*Status*
Updated Indication	■ Green	UPDATED = 1	*Updated*
Updated text	**Updated** Black		*Text behind the indication*
Manual entry Indication	■ Green	MAN_ENTR = 1	*Manual entry flag*
Manual entry text	**Man Entry** Black		*Text behind the indication*
Alarm period block Indication	■ Yellow	AL_P_BLK = 1	*Alarm blocked by PC-program*
Alarm period block text	**Alarm Period Blk** Black		*Text behind the indication*
Disturbance Indication	■ Red	DISTURB = 1	*Fault*
Disturbance text	**Disturbance** Black		*Text behind the indication*
Signal error Indication	■ Red	ERR = 1	*I/O Error*
Signal error text	**Signal Error** Black		*Text behind the indication*
Overflow Indication	■ Red	OVERFLOW = 1	*Overflow*
Overflow text	**Overflow** Black		*Text behind the indication*
Unacknowledged Indication	■ Red	ERR = 1	*Signal error*
Unacknowledged text	**Unacknowledged** Black		*Text behind the indication*
PARAMETERS			*Parameters*
Implemented Indication	■ Green	ACT = 1	*Implemented*
Implemented text	**Implemented** Black		*Text behind the indication*
Normal treat Indication	■ Green	NORM_TR = 1	*Normal object treat*
Normal treat text	**Normal Treated** Black		*Text behind the indication*
No overflow alarms Indication	■ Green	OVF_CTRL = 1	*No error at overflow*
No overflow alarms text	**No Overflow Alarms** Black		*Text behind the indication*
Tested Indication	■ Green	TESTED = 1	*Tested*
Tested text	**Tested** Black		*Text behind the indication*
Input Blocked text	**Input Blocked** Yellow	BLOCKED = 1	*Input blocked*

Description	Presentation	Condition	Remarks
	-	BLOCKED = 0	*Invisible*
Value text	**Value** Black		*Text in front of the value*
Value value	Red	ERR = 1	*I/O Error*
	Green	VALUE	*Measured value*
Value unit	Green	UNIT	*Measured unit*
Range Max text	**Range Max** Black		*Text in front of the value*
Range Max		RANGEMAX	*Range Max of the MV value*
	Green	-	
Range Min text	**Range Min** Black		*Text in front of the value*
Range Min		RANGEMAX	*Range Min of the MV value*
	Green	-	

C.2 AO Object Presentation

The AO Object Presentation presents information for Analog Output objects, Figure 3-2.



AOC71_1

Section 5 Class 2

OUT H 80.00

L 15.00

☐ Man

Auto

Value 58.00 %

Range Max 100.00

Range Min .00

Count: *1

Figure 3-2. AO Object Presentation

The operation of this presentation is described in the following table.

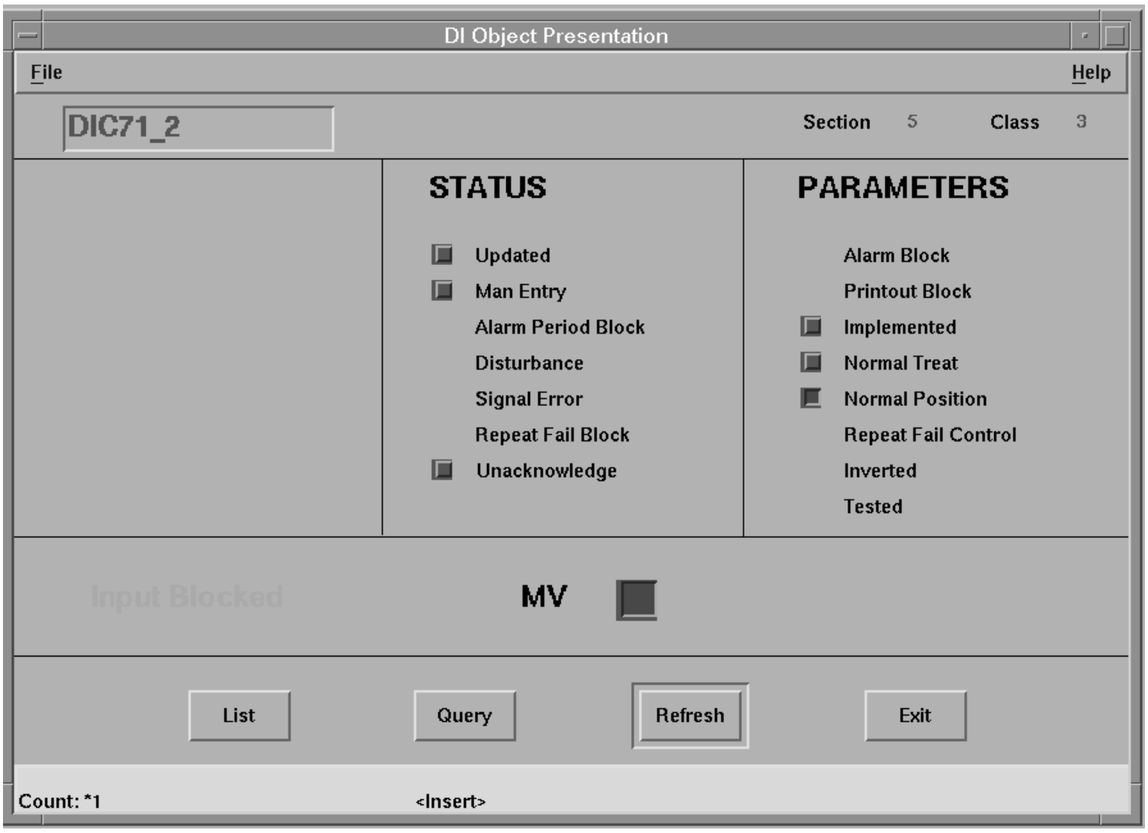
Description	Presentation	Condition	Remarks
HEADER			*Header*
Name field		NAME	*Object Name*
	Yellow	OUTP_BLK = 1	*Output blocked*
	Red	ERR = 1	*Alarm*
	Cyan	MAN = 1	*Manual mode*
	Magenta	-	*Normal*
Description			*Description*
	Yellow	OUTP_BLK = 1	*Output blocked*
	Red	ERR = 1	*Alarm*

Description	Presentation	Condition	Remarks
	Cyan	MAN = 1	*Manual mode*
	Magenta	-	*Normal*
Section text	**Section** Black		*Text in front of the value*
Section value	Green	PROC_SEC	*Process section*
Class text	**Class** Black		*Text in front of the value*
Class value	Green	CLASS	*Object class*
ALARM LIMITS			*Alarm limits*
Out text	**OUT** Black		
High Limitation Indication	■ Yellow	VALUE_HI = 1	*High limitation flag*
H text	**H** Black		*Text in front of the value*
H value		HI_LIM	*High limitation value*
	-	EN_HI = 1 and VALUE_HI = 1	*Invisible*
	Cyan	MAN = 1	*Manual mode*
	Magenta	-	*Auto mode*
Low Limitation Indication	■ Yellow	VALUE_LO = 1	*Low limitation flag*
L text	**L** Black		*Text in front of the value*
L value		LO_LIM	*Low limitation value*
	-	LO_LIM < RANGEMIN or EN_LO = 0	*Invisible*
	Cyan	MAN = 1	*Manual mode*
	Magenta	-	*Auto mode*
STATUS			*Status*
Signal Error Indication	■ Red	ERR = 1	*Signal error*
Signal Error text	**Signal Error** Black		*Text behind the indication*
Unacknowledged Indication	■ Red	ERR = 1	*Signal error*
Unacknowledged text	**Unacknowledged** Black		*Text behind the indication*
PARAMETERS			*Parameters*
Implemented Indication	■ Green	IMPLEMENTED = 1	*Implemented*
Implemented text	**Implemented** Black		*Text behind the indication*
Use Max Limitation Indication	■ Green	EN_HI = 1	*Use max limitation*
Use Max Lim text	**Use Max Lim** Black		*Text behind the indication*
Use Min Limitation Indication	■ Green	EN_LO = 1	*Use min limitation*

Description	Presentation	Condition	Remarks
Use Min Lim text	**Use Min Lim** Black		*Text behind the indication*
Output Restart Indication	■ Green	RESTART = 1	*Output restart*
Output Restart text	**Output Restart** Black		*Text behind the indication*
Tested Indication	■ Green	TESTED = 1	*Tested*
Tested text	**Tested** Black		*Text behind the indication*
Man Indication	■ Cyan	MAN = 1	*Manual mode*
Man text	**Man**		*Text behind the indication*
	Cyan	MAN = 1	*Manual mode*
	Black	MAN = 0	*Auto mode*
Auto Indication	■ Magenta	MAN = 0	*Auto mode*
Auto text	**Auto**		*Text behind the indication*
	Magenta	MAN = 0	*Auto mode*
	Black	MAN = 1	*Manual mode*
Output Blocked	**Output Blocked** Yellow	OUTP_BLK=1	*Output Blocked*
	-	OUTP_BLK=0	*Invisible*
Value text	**Value** Black		*Text in front of the value*
Value value		VALUE	
	Cyan	MAN = 1	*Manual mode*
	Magenta	-	*Auto mode*
Value unit		UNIT	
	Cyan	MAN = 1	*Manual mode*
	Magenta	-	*Auto mode*
Range Max text	**Range Max** Black		*Text in front of the value*
Range Max		RANGEMAX	*Range max of the Output value*
	Cyan	MAN = 1	*Manual mode*
	Magenta		*Auto mode*
Range Min text	**Range Min** Black		*Text in front of the value*
Range Min		RANGEMIN	*Range min of the Output value*
	Cyan	MAN = 1	*Manual mode*
	Magenta		*Auto mode*

C.3 DI Object Presentation

The DI Object Presentation presents information for selected Digital Input objects, Figure 3-3.



DIC71_2

Section 5 Class 3

STATUS

- ☑ Updated
- ☑ Man Entry
- Alarm Period Block
- Disturbance
- Signal Error
- Repeat Fail Block
- ☑ Unacknowledge

PARAMETERS

- Alarm Block
- Printout Block
- ☑ Implemented
- ☑ Normal Treat
- ☑ Normal Position
- Repeat Fail Control
- Inverted
- Tested

Input Blocked

MV ☑

Count: *1 <Insert>

Figure 3-3. DI Object Presentation

The operation of this presentation is described in the following table.

Description	Presentation	Condition	Remarks
HEADER			*Header*
Name field		NAME	*Object Name*
	Yellow	BLOCKED = 1	*Input blocked*
	Red	DISTURB = 1 or ERR = 1	*Alarm*
	Green	-	*Normal*
Description		DESCR	*Description*
	Yellow	BLOCKED = 1	*Input blocked*

Description	Presentation	Condition	Remarks
	Red	DISTURB = 1 or ERR = 1	*Alarm*
	Green	-	*Normal*
Section text	**Section** Black		*Text in front of the value*
Section value	Green	PROC_SEC	*Process section*
Class text	**Class** Black		*Text in front of the value*
Class value	Green	CLASS	*Object class*
STATUS			*Status*
Updated indication	■ Green	UPDATED = 1	*Updated*
Updated text	**Updated** Black		*Text behind the indication*
Manual entry indication	■ Green	MAN_ENTR = 1	*Manual entry flag*
Manual entry text	**Man Entry** Black		*Text behind the indication*
Alarm period block indication	■ Yellow	AL_P_BLK = 1	*Alarm blocked by PC-program*
Alarm period block text	**Alarm Period Blk** Black		*Text behind the indication*
Disturbance indication	■ Red	DISTURB = 1	*Fault*
Disturbance text	**Disturbance** Black		*Text behind the indication*
Signal error indication	■ Red	ERR = 1	*I/O Error*
Signal error text	**Signal Error** Black		*Text behind the indication*
Repeat fail block indication	■ Yellow	RP_F_BLK = 1	*Repeat fail blocked*
Repeat fail block text	**Repeat Fail Block** Black		*Text behind the indication*
Unacknowledged indication	■ Red	ERR = 1	*Signal error*
Unacknowledged text	**Unacknowledged** Black		*Text behind the indication*
PARAMETERS			*Parameters*
Alarm block indication	■ Yellow	**Alarm Block** = 1	*Alarm blocked by operator*
Alarm block text	**Alarm Block** Black		*Text behind the indication*
Print Block indication	■ Yellow	PR_BLK = 1	*Print blocked*
Print Block text	**Print Block** Black		*Text behind the indication*
Implemented indication	■ Green	ACT = 1	*Implemented*
Implemented text	**Implemented** Black		*Text behind the indication*
Normal treat indication	■ Green	NORM_TR = 1	*Normal object treat*
Normal treat text	**Normal Treat** Black		*Text behind the indication*
Normal position indication			*Normal position*

Description	Presentation	Condition	Remarks
	■ Green	NORM_POS = 1	*Symbol released*
	□ Green	NORM_POS = 0	*Symbol pressed in*
Normal position text	**Normal Position** Black		*Text behind the indication*
Repeat Fail Control indication	■ Green	RP_F_CTL = 1	*Repeat fail control*
Repeat Fail Control text	**Repeat Fail Control** Black		*Text behind the indication*
Inverted indication	■ Green	INV = 1	*Inverted*
Inverted text	**Inverted** Black		*Text behind the indication*
Tested indication	■ Green	TESTED = 1	*Tested*
Tested text	**Tested** Black		*Text behind the indication*
Input blocked	**Input Blocked** Yellow	BLOCKED=1	*Input Blocked*
	-	BLOCKED=0	*Invisible*
MV text	**MV** Black		*Text in front of the value*
MV value			*3D Display object*
	■ Green	VALUE = 1	*Symbol released*
	□ Green	VALUE = 0	*Symbol pressed in*

C.4 DO Object Presentation

DO Object Presentation presents information for selected Digital Output object, Figure 3-4.

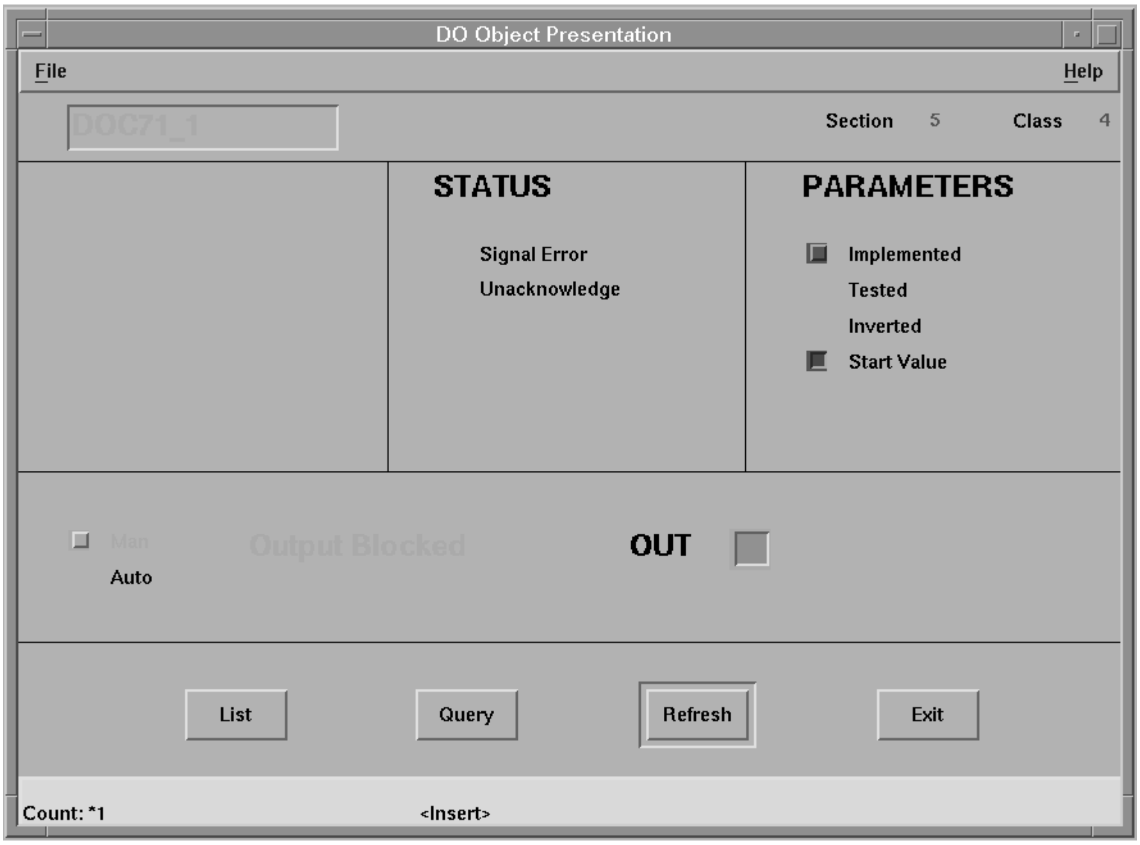


Figure 3-4. DO Object Presentation

The operation of this presentation is described in the following table.

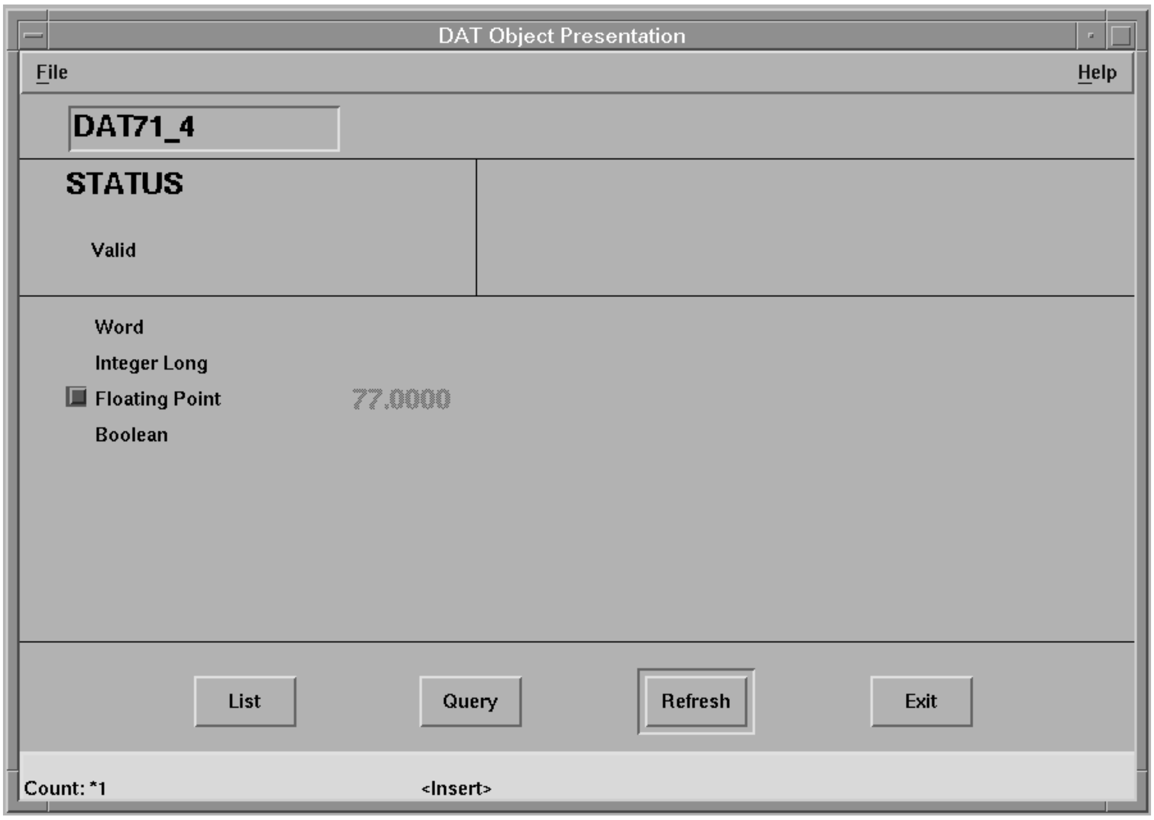
Description	Presentation	Condition	Remarks
HEADER			*Header*
Name field		NAME	*Object Name*
	Yellow	OUTP_BLK = 1	*Output blocked*
	Red	ERR = 1	*Alarm*
	Cyan	MAN = 1	*Manual mode*
	Magenta	-	*Normal*
Description		DESCR	*Description*
	Yellow	OUTP_BLK = 1	*Output blocked*

Description	**Presentation**	**Condition**	**Remarks**
	Red	ERR = 1	*Alarm*
	Cyan	MAN = 1	*Manual mode*
	Magenta	-	*Normal*
Section text	**Section** Black		*Text in front of the value*
Section value	Green	PROC_SEC	*Process section*
Class text	**Class** Black		*Text in front of the value*
Class value	Green	CLASS	*Object class*
STATUS			*Status*
Signal error Indication	■ Red	ERR = 1	*I/O Error*
Signal error text	**Signal Error** Black		*Text behind the indication*
Unacknowledged Indication	■ Red	ERR = 1	*Signal error*
Unacknowledged text	**Unacknowledged** Black		*Text behind the indication*
PARAMETERS			*Parameters*
Implemented Indication	■ Green	ACT = 1	*Implemented*
Implemented text	**Implemented** Black		*Text behind the indication*
Tested Indication	■ Green	TESTED = 1	*Tested*
Tested text	**Tested** Black		*Text behind the indication*
Inverted Indication	■ Green	INV = 1	*Inverted*
Inverted text	**Inverted** Black		*Text behind the indication*
Start Value Indication			*Start value*
	Last Val Green	RESTART = 1	*Text indication*
	■ Green	STVAL = 1 and RESTART = 0	*Symbol released*
	□ Green	STVAL = 0 and RESTART = 0	*Symbol pressed in*
Start Value text	**Start Value** Black		*Text behind the indication*
Man Indication	■ Cyan	MAN = 1	*Manual mode*
Man text	**Man**		*Text behind the indication*
	Cyan	MAN = 1	*Manual mode*
	Black	MAN = 0	*Auto mode*
Auto Indication	■ Magenta	MAN = 0	*Auto mode*
Auto text	**Auto**		*Text behind the indication*
	Magenta	MAN = 0	*Auto mode*
	Black	MAN = 1	*Manual mode*

Description	Presentation	Condition	Remarks
Output Blocked	**Output Blocked** Yellow	OUTP_BLK = 1	*Output blocked*
	-	OUTP_BLK = 0	*Invisible*
OUT text	**OUT** Black		*Text in front of the value*
Value			*3D Display object*
	■ Cyan	MAN = 0 and VALUE = 1	*Symbol released*
	□ Cyan	MAN = 1 and VALUE = 0	*Symbol pressed in*
	■ Magenta	VALUE = 1	*Symbol released*
	□ Magenta	VALUE = 0	*Symbol pressed in*

C.5 DAT Object Presentation

The DAT Object Presentation presents information for selected DAT object, Figure 3-5.



DAT71_4

STATUS

Valid

Word

Integer Long

■ Floating Point 77.0000

Boolean

Count: *1 <Insert>

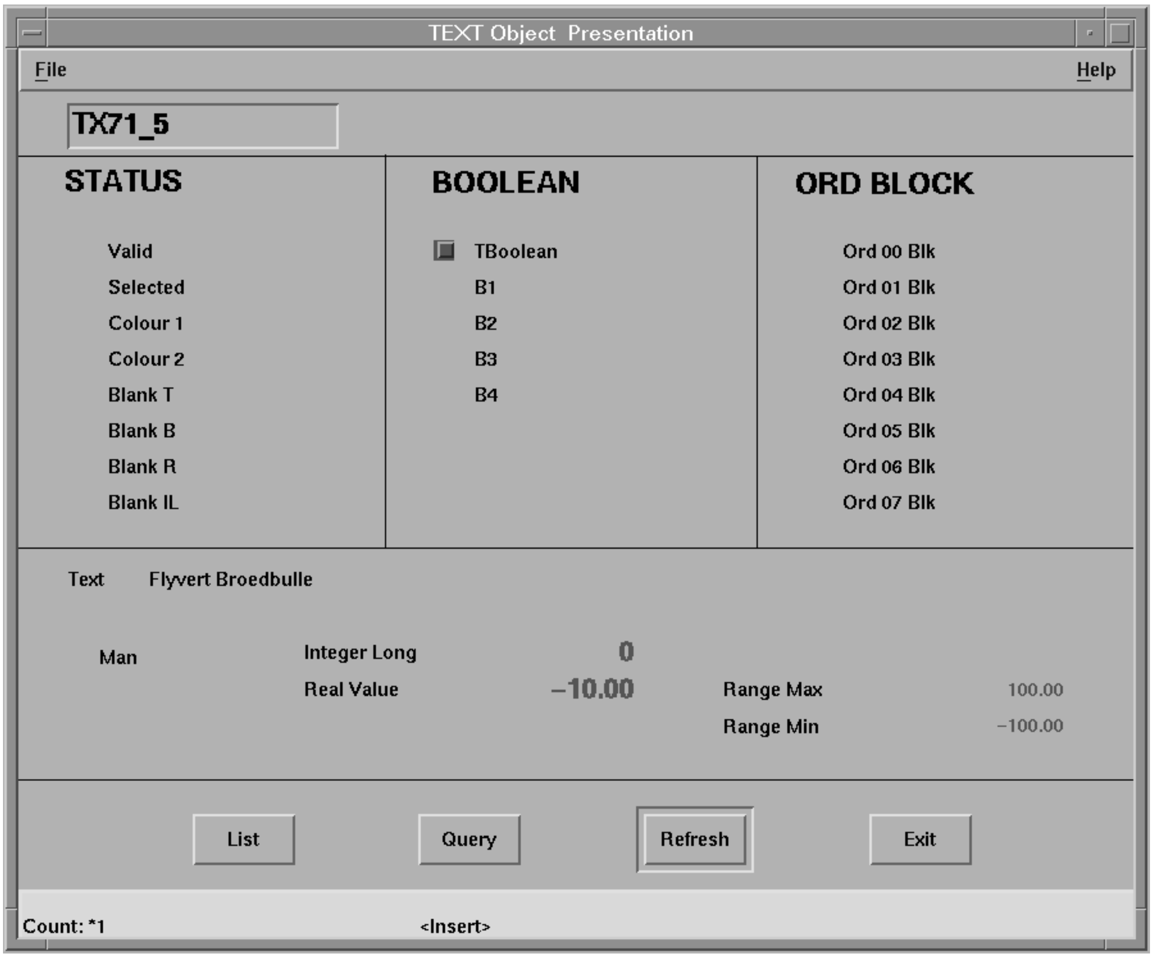
Figure 3-5. DAT Object Presentation

The operation of this presentation is described in the following table..

Description	**Presentation**	**Condition**	**Remarks**
Object name field	Black		*Name of the DAT object*
STATUS			*Status*
Valid Indication	■ Green		*Valid*
Valid text	**Valid** Black		*Text behind the indication*
Integer Word Indication	■ Green		*Integer word type*
Integer Word text	**Integer Word** Black		*Text behind the indication*
Integer Word value	Green		*Integer word value*
Integer Long Indication	■ Green		*Integer long type*
Integer Long text	**Integer Long** Black		*Text behind the indication*
Integer Long value	Green		*Integer long type*
Floating Point Indication	■ Green		*Floating point type*
Floating Point text	**Floating Point** Black		*Text behind the indication*
Floating Point value	Green		*Floating point value*
Boolean Indication	■ Green		*Boolean type*
Boolean text	**Boolean** Black		*Text behind the indication*
B0 Indication	■ Green		*Boolean value 0*
B0 text	**B0** Black		*Text behind the indication*
B1 Indication	■ Green		*Boolean value 1*
B1 text	**B1** Black		*Text behind the indication*
B2 Indication	■ Green		*Boolean value 2*
B2 text	**B2** Black		*Text behind the indication*
B3 Indication	■ Green		*Boolean value 3*
B3 text	**B3** Black		*Text behind the indication*
:	:		*:*
:	:		*:*
B31 Indication	■ Green		*Boolean value 31*
B31 text	**B31** Black		*Text behind the indication*

C.6 TEXT Object Presentation

The TEXT Object Presentation presents information for the selected TEXT object, Figure 3-6.



TX71_5

Text Flyvert Broedbulle

Integer Long 0

Real Value −10.00

Range Max 100.00

Range Min −100.00

Count: *1 <Insert>

Figure 3-6. TEXT Object Presentation

The operation of this presentation is described in the following table.

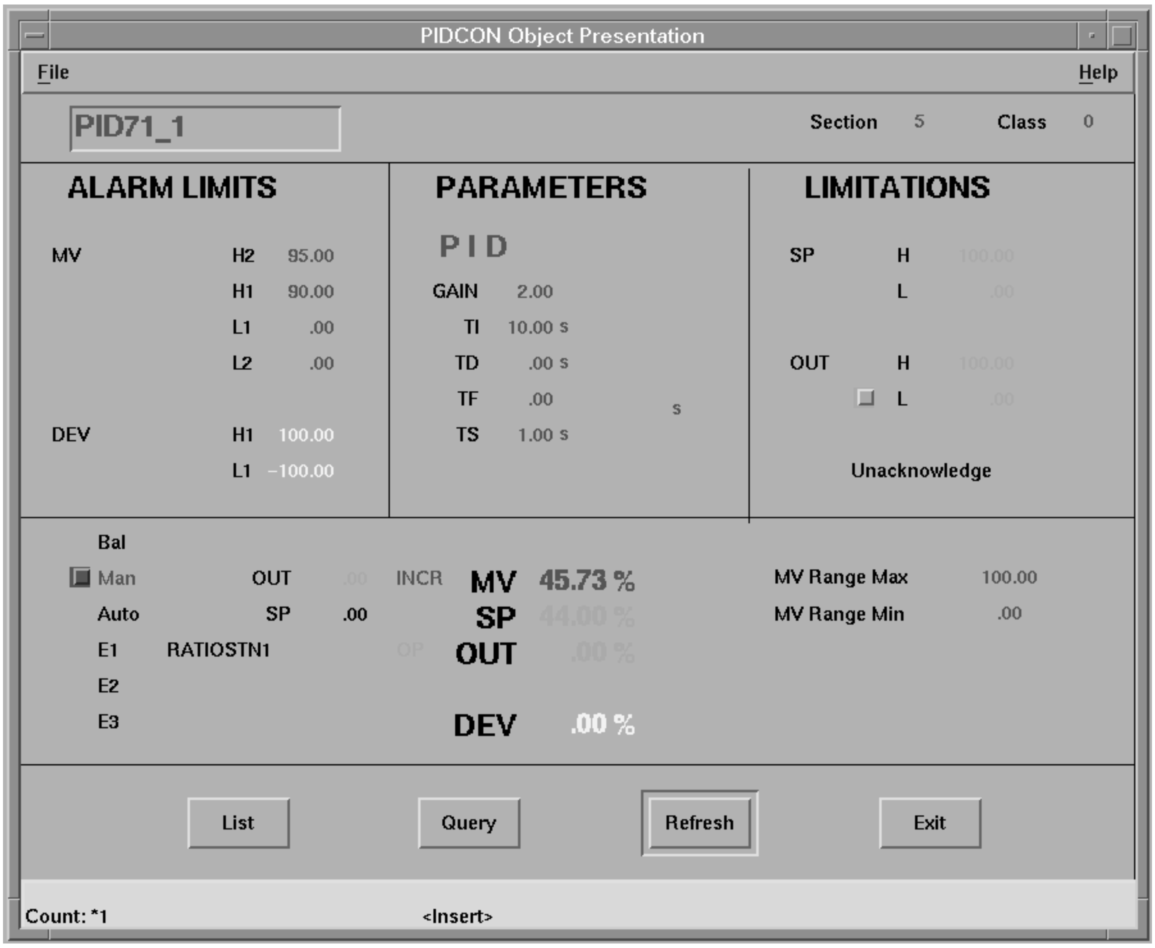
Description	Presentation	Condition	Remarks
Object name field	Black		*Name of the TEXT object*
STATUS			*Status*
Valid Indication	■ Green		*Object is updated since restart*
Valid text	**Valid** Black		*Text behind the indication*
Selected Indication	■ Green		*Object is selected by an operator*
Selected text	**Selected** Black		*Text behind the indication*

Description	Presentation	Condition	Remarks
Color 1 Indication	■ Green		*Not used in AdvaInform Basic Object*
Color 1 text	**Color 1** Black		*Text behind the indication*
Color 2 Indication	■ Green		*Not used in AdvaInform Basic Object*
Color 2 text	**Color 2** Black		*Text behind the indication*
Blank T Indication	■ Green		
Blank T text	**Blank T** Black		*Text behind the indication*
Blank B Indication	■ Green		
Blank B text	**Blank B** Black		*Text behind the indication*
Blank R Indication	■ Green		
Blank R text	**Blank R** Black		*Text behind the indication*
Blank IL Indication	■ Green		
Blank IL text	**Blank IL** Black		*Text behind the indication*
BOOLEAN			*Boolean*
TBoolean Indication	■ Green		*Boolean value*
TBoolean text	**TBoolean** Black		*Text behind the indication*
B1 Indication	■ Green		*Boolean value*
B1 text	**B1** Black		*Text behind the indication*
B2 Indication	■ Green		*Boolean value*
B2 text	**B2** Black		*Text behind the indication*
B3 Indication	■ Green		*Boolean value*
B3 text	**B3** Black		*Text behind the indication*
B4 Indication	■ Green		*Boolean value*
B4 text	**B4** Black		*Text behind the indication*
ORD BLOCK			*Order Blocks*
Ord 00 Blk Indication	■ Yellow		*Blocking Text value*
Ord 00 Blk text	**Ord 00 Blk** Black		*Text behind the indication*
Ord 01 Blk Indication	■ Yellow		*Blocking Boolean value*
Ord 01 Blk text	**Ord 01 Blk** Black		*Text behind the indication*
Ord 02 Blk Indication	■ Yellow		*Blocking Real value*
Ord 02 Blk text	**Ord 02 Blk** Black		*Text behind the indication*
Ord 03 Blk Indication	■ Yellow		*Blocking INT_LONG value*
Ord 03 Blk text	**Ord 03 Blk** Black		*Text behind the indication*

Description	Presentation	Condition	Remarks
Ord 04 Blk Indication	■ Yellow		*Not used*
Ord 04 Blk text	**Ord 04 Blk** Black		*Text behind the indication*
Ord 05 Blk Indication	■ Yellow		*Not used*
Ord 05 Blk text	**Ord 05 Blk** Black		*Text behind the indication*
Ord 06 Blk Indication	■ Yellow		*Not used*
Ord 06 Blk text	**Ord 06 Blk** Black		*Text behind the indication*
Ord 07 Blk Indication	■ Yellow		*Not used*
Ord 07 Blk text	**Ord 07 Blk** Black		*Text behind the indication*
Text text	**Text** Black		*Text in front of field*
Text field	Black		*Text string*
Man Indication	■ Cyan		*Manual mode*
Man text	**Man** Black		*Text behind the indication*
Integer Long text	**Integer long** Black		*Text in front of value*
Integer Long value	Green		*Integer long value*
Real Value text	**Real Value** Black		*Text in front of value*
Real Value value	Green		*Real value*
Range Max text	**Range Max** Black		*Text in front of value*
Range Max value	Green		*Maximum value to be entered for Text_Real*
Range Min text	**Range Min** Black		*Text in front of value*
Range Min value	Green		*Minimum value to be entered for Text_Real*

C.7 PIDCON Object Presentation (Master only)

The PIDCON Object Presentation presents information for the selected PIDCON object, see Figure 3-7. PIDCON is the PID controller object type for the Advant OCS with Master software.



PID71_1

Section 5
Class 0

MV H2 95.00
H1 90.00
L1 .00
L2 .00
DEV H1 100.00
L1 -100.00

PID
GAIN 2.00
TI 10.00 s
TD .00 s
TF .00
TS 1.00 s
s

SP H 100.00
L .00
OUT H 100.00
☐ L .00

Bal
■ Man
Auto
E1 RATIOSTN1
E2
E3

OUT .00
SP .00
INCR
OP

MV 45.73 %
SP 44.00 %
OUT .00 %
DEV .00 %

MV Range Max 100.00
MV Range Min .00

Count: *1
<Insert>

Figure 3-7. PIDCON Object Presentation

The operation of this presentation is described in the following table.

Description	Presentation	Condition	Remarks
HEADER			*Header*
Name		NAME	*Object name*
	Red	DISTURB = 1	*Alarm*
	Green	-	*Normal*
Description		DESCR	*Description*
	Red	DISTURB = 1	*Alarm*
	Green	-	*Normal*
Section	**Section** Black		*Text in front of the value*
	Green	PROC_SEC	*Process section*
Class	**Class** Black		*Text in front of the value*
	Green	CLASS	*Object class*
ALARM LIMITS			*Alarm limits*
MV text	**MV** Black		*Text in front of the line*
Printout blocked	**P** Yellow	PRINT_F2_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F2_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_F2_P_BLK = 1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV>H2 = 1	*Unacknowledged alarm*
	■ Red	MV>H2 = 1	*Alarm*
H2 text	**H2** Black		*Text in front of the value*
H2 value	Green	MVH2	*High limit 2 value*
Printout blocked	**P** Yellow	PRINT_F1_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F1_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_F1_P_BLK = 1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV>H1 = 1	*Unacknowledged alarm*
	■ Red	MV>H1 = 1	*Alarm*
H1 text	**H1** Black		*Text in front of the value*
H1 value	Green	MVH1	*High limit 1 value*
Printout blocked	**P** Yellow	PRINT_F1_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F1_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_F1_P_BLK = 1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV<L1 = 1	*Unacknowledged alarm*
	■ Red	MV<L1 = 1	*Alarm*

Description	Presentation	Condition	Remarks
L1 text	**L1** Black		*Text in front of the value*
L1 value	Green	MVL1	*Low limit 1 value*
Printout blocked	**P** Yellow	PRINT_F2_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F2_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_F2_P_BLK = 1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV<L2 = 1	*Unacknowledged alarm*
	■ Red	MV<L2 = 1	*Alarm*
L2 text	**L2** Black		*Text in front of the value*
L2 value	Green	MVL2	*Low limit 2 value*
DEV	**DEV** Black		
Printout blocked	**P** Yellow	PRINT_F1_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F1_BLK =1	*Alarm block by operator*
	Bx Yellow	AL_F1_P_BLK = 1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_DEV>H1 = 1	*Unacknowledged alarm*
	■ Red	DEV>H1 = 1	*Alarm*
H text	**H** Black		*Text in front of the value*
H value	Cream	DEVH1	
Printout blocked	**P** Yellow	PRINT_F1_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F1_BLK =1	*Alarm blocked by operator*
	Bx Yellow	AL_F1_P_BLK = 1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_DEV<L1 = 1	*Unacknowledged alarm*
	■ Red	DEV<L1 = 1	*Alarm*
L text	**L** Black		*Text in front of the value*
L value	Cream	DEVL1	
PARAMETERS	Black		
P	**P** Green		*Gain active*
I	**I** Green	BLK_I = 0 and EINTBE=0	*Integration active*
D	**D** Green	BLK_D = 0 and EDERBE = 0	*Derivation active*
Gain text	**Gain** Black		*Gain*
Gain value	Black	EGAINE = 1	*Value changed by operator*
	Green	GAIN	
External value	Green	EGAINE = 1	

Description	**Presentation**	**Condition**	**Remarks**
TI text	**TI** Black		*Integration time*
TI value	Black	ETIE = 1	*Value changed by operator*
	Green	TI	
TI unit	**S** Green		
External value	Green	ETIE = 1	
External unit	**S** Green	ETIE = 1	
TD text	**TD** Black		*Derivation time*
TD value	Black	ETDE = 1	*Value changed by operator*
	Green	TD	
TD unit	**S** Green		
External value	Green	ETDE = 1	
External unit	**S** Green	ETDE = 1	
TF text	**TF** Black		*Filter Time*
TF value	Black	ETFE = 1	*Value changed by operator*
	Green	TF	
TF unit	**S** Green		
External value	Green	ETFE = 1	
External unit	**S** Green	ETFE = 1	
TS text	**TS** Black		*Sampling time*
TS Value	Green	TS	
TS unit	**S** Green		
DZ text	**DZ** Black		*Deadzone*
DZ value	Green	DZ, CTRL = ON-OFF	
DZ unit	**%** Green	DZ, CTRL = ON-OFF	
LIMITATIONS			*Limitations*
SP text	**SP** Black		*Setpoint*
Warning indication	■ Yellow	SP=HL = 1	
H text	**H** Black		*High limit setpoint*
H value	Cyan	SETPH	
Warning indication	■ Yellow	SP=LL = 1	
L text	**L** Black		*Low limit setpoint*
L value	Cyan	SETPL	
OUT text	**OUT** Black		*Text in front of the line*

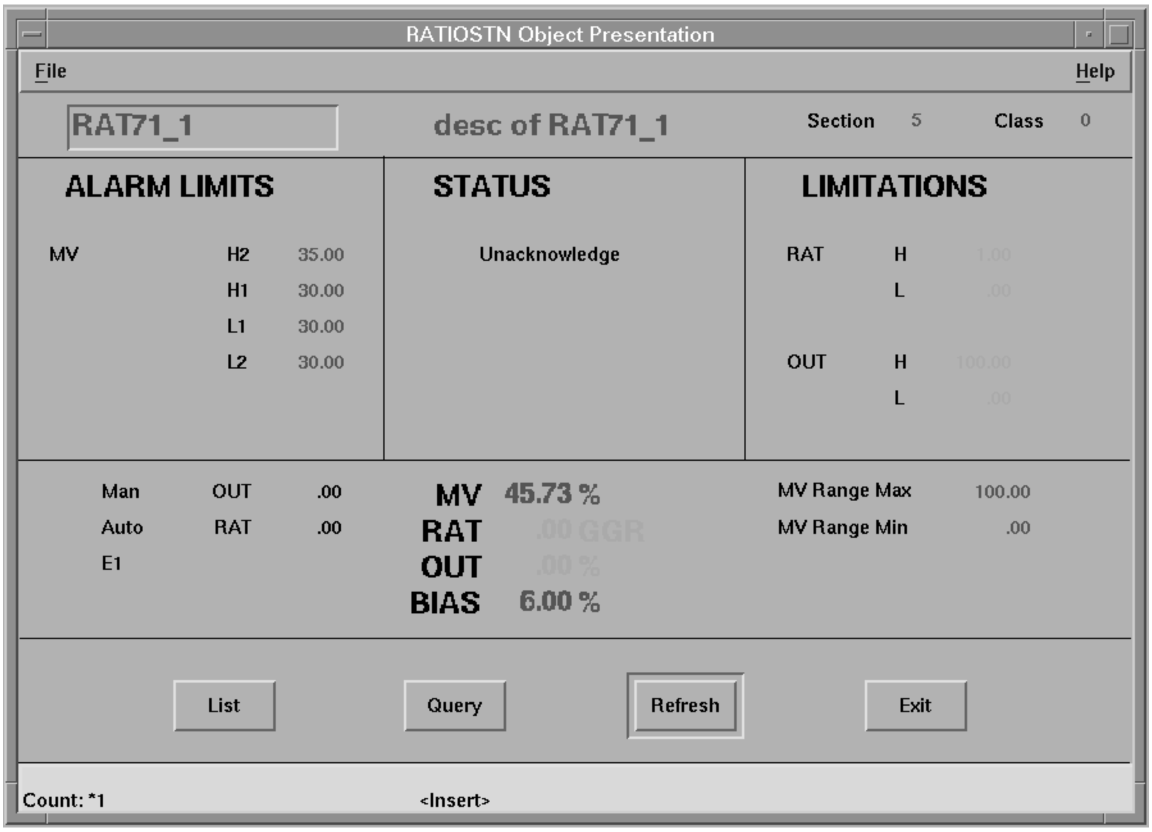
Description	Presentation	Condition	Remarks
Warning indication	■ Yellow	OUT_HL = 1	
H text	**H** Black		*High limit output*
H value	Black	OUTPH, EOLIM = 1	*Output High value*
	Magenta	OUTPH, EOLIM = 0	
External H value	Magenta	EOLIM = 1	*External Output High value*
	Black		
Warning Indication	■ Yellow	OUT_EQ_LL = 1	
L text	**L** Black		*Low limit output*
L value	Black	OUTPL, EOLIM = 1	*Output Low value*
	Magenta	OUTPL, EOLIM = 0	
External L value	Magenta	EOLIM = 1	*External Output Low value*
	Black	-	
Unacknowledge indication	■ Red	AL_UNACK = 1	*Unacknowledged alarm*
Unacknowledge text	**Unacknowledge** Black		*Text behind the indication*
Bal indication	■ Green	BAL = 1	*Bal control mode*
Bal mode	**Bal** Green	BAL = 1	*Text behind the indication*
	Bal Black	-	
	Local Green	LOCAL = 1	*Bal Local control mode*
Man indication	■ Green	MAN = 1	*Man control mode*
Man mode	**Man** Green	MAN = 1	*Text behind the indication*
	Man Black	-	
	Clamped Green	CLAMP = 1	*Man Clamped control mode*
	Forced Green	MANFD = 1	*Man Forced control mode*
Man Out text	**OUT** Black		*Text in front of the value*
Man Out value		MAN_OUT	*Man Out value*
	Magenta	MAN = 1	*Manual mode*
	Black	-	
Auto indication	■ Green	AUTO = 1	*Auto control mode*
Auto mode	**Auto** Green	AUTO = 1	*Text behind the indication*
	Auto Black	-	
Auto Setpoint text	**SP** Black		*Text in front of the value*
Auto Setpoint value		AUTOSP	*Auto Setpoint value*
	Cyan	AUTO =1	*Auto mode*

Description	**Presentation**	**Condition**	**Remarks**
	Black	-	
E1 indication	■ Green	E1 = 1	*E1 control mode*
E1 mode	**E1** Green	E1 = 1	*Text behind the indication*
	E1 Black	-	
E1 name		E1NAME	*User text behind the text E1*
	Green	E1 = 1	
	Black	-	
E2 indication	■ Green	E2 = 1	*E2 control mode*
E2 mode	**E2** Green	E2 = 1	*Text behind the indication*
	E2 Black	-	
E2 name		E2NAME	*User text behind the text E2*
	Green	E2 = 1	
	Black	-	
E3 indication	■ Green	E3 = 1	*E3 control mode*
E3 mode	**E3** Green	E3 = 1	*Text behind the indication*
	E3 Black	-	
E3 name		E3NAME	*User text behind the text E3*
	Green	E3 = 1	
	Black	-	
MV text	**MV** Black		*Text in front of the value*
MV value	Red	AI_ERR = 1	*Signal error*
		MV	*Measured value*
	Green	-	
MV unit		UNIT	*Measured unit*
	Green	-	
SP text	**SP** Black		*Text in front of the value*
SP value		SETP	*Setpoint value*
	Cyan	-	
SP unit		UNIT	*Setpoint unit*
	Cyan	-	
OUT text	**OUT** Black		*Text in front of the value*
OUT value		POUT	*Output value*
	Magenta	-	

Description	Presentation	Condition	Remarks
OUT unit	**%** Magenta		*Output unit*
External OUT Limitation indication	**X** Yellow	(OUT=HL = 1 or OUT=LL= 1) and EOLIM = 1	*OUT limited by external limitation*
ACT text	**ACT** Black	CTRL = ON-OFF	*Text in front of the value*
ACT value		ACTPOS	*Actuator value*
	Blue	CTRL = ON-OFF	
ACT unit	**%** Blue	CTRL = ON-OFF	*Actuator unit*
DEV text	**DEV** Black		*Text in front of the value*
DEV value		DEV	*Deviation value*
	Cream	-	
DEV unit	**%** Cream		*Deviation unit*
MV Range Max		MAX	*Range Max of the MV and SP value*
	Green	-	
OUT Range Max			*Range Max of OUT is 100.0 %*
	100.0 Magenta	-	
MV Range Min		MIN	*Range Min of the MV and SP value*
	Green	-	
OUT Range Min			*Range Min of OUT is 0.0 %*
	0.0 Magenta	-	

C.8 RATIOSTN Object Presentation (Master only)

The RATIOSTN Object Presentation presents information for the selected RATIONSTN object, see Figure 3-8. RATIOSTN is used when the ratio between two or more process values must be controlled. The output is usually used as input to another controller function.



RAT71_1
desc of RAT71_1
Section 5
Class 0

ALARM LIMITS
MV H2 35.00
H1 30.00
L1 30.00
L2 30.00

STATUS
Unacknowledge

LIMITATIONS
RAT H 1.00
L .00
OUT H 100.00
L .00

Man OUT .00
Auto RAT .00
E1

MV 45.73 %
RAT .00 GGR
OUT .00 %
BIAS 6.00 %

MV Range Max 100.00
MV Range Min .00

Count: *1
<Insert>

Figure 3-8. RATIOSTN Object Presentation

The operation of this presentation is described in the following table.

Description	Presentation	Condition	Remarks
HEADER			*Header*
Name		NAME	*Object name*
	Red	DISTURB = 1	*Alarm*
	Green	-	*Normal*
Description		DESCR	*Description*
	Red	DISTURB = 1	*Alarm*
	Green	-	*Normal*

Description	Presentation	Condition	Remarks
Section text	**Section** Black		*Text in front of the value*
Section value	Green	PROC_SEC	*Process section*
Class text	**Class** Black		*Text in front of the value*
Class value	Green	CLASS	*Object class*
ALARM LIMITS			*Alarm limits*
MV text	**MV** Black		*Text in front of the line*
Printout blocked	**P** Yellow	PRINT_F2_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F2_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_F2_P_BLK = 1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV>H2 = 1	*Alarm High limit 2*
	■ Red	MV> H2 = 1	
H2 text	**H2** Black		*Text in front of the value*
H2 value		MVH2	*High limit 2 value*
	Green	-	
Printout blocked	**P** Yellow	PRINT_F1_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F1_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_F1_P_BLK = 1	*Alarm blocked by PC-program*
Warning indication	■ Red flashing	AU_MV>H1 = 1	*Warning High limit 1*
	■ Red	MV> H1 = 1	
H1 text	**H1** Black		*Text in front of the value*
H1 value		MVH1	*High limit 1 value*
	Green	-	
Printout blocked	**P** Yellow	PRINT_F1_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F1_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_F1_P_BLK = 1	*Alarm blocked by PC-program*
Warning indication	■ Red flashing	AU_MV<L1 = 1	*Warning Low limit 1*
	■ Red	MV<L1 = 1	
L1 text	**L1** Black		*Text in front of the value*
L1 value		MVL1	*Low limit 1 value*
	Green	-	
Printout blocked	**P** Yellow	PRINT_F2_BLK = 1	*Printout blocked*
Alarm blocked	**B** Yellow	AL_F2_BLK = 1	*Alarm blocked by operator*
	Bx Yellow	AL_F2_P_BLK = 1	*Alarm blocked by PC-program*

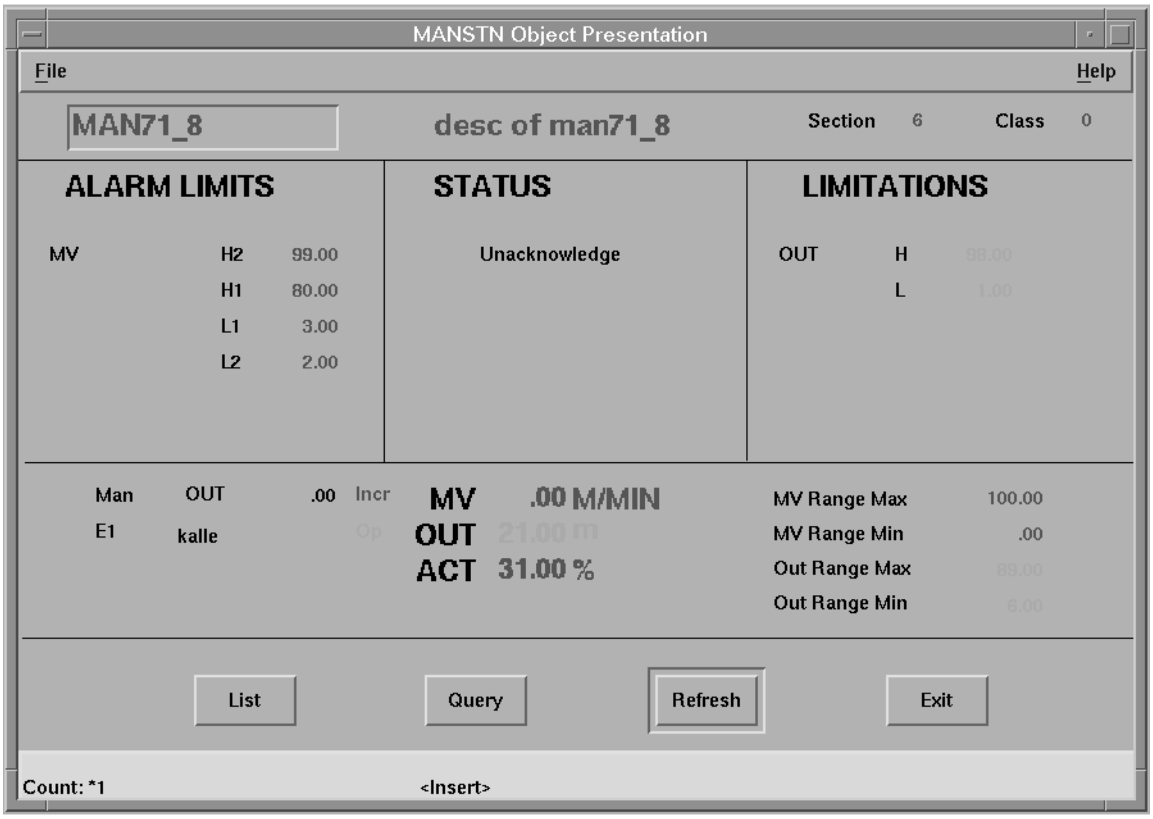
Description	Presentation	Condition	Remarks
Alarm indication	■ Red flashing	AU_MV<L2 = 1	*Alarm Low limit 2*
	■ Red	MV<L2 = 1	
L2 text	**L2** Black		*Text in front of the value*
L2 value		MVL2	*Low limit 2 value*
	Green	-	
STATUS			*Status*
Unacknowledge indication	■ Red	AL_UNACK = 1	*Unacknowledged alarm*
Unacknowledge text	**Unacknowledge** Black		*Text behind the indication*
LIMITATIONS			*Limitations*
RAT text	**RAT** Black		*Text in front of the line*
Warning indication	■ Yellow	RATIO=HL = 1	*Warning Ratio High*
H text	**H** Black		*Text in front of the value*
H value		RATIOH	*Ratio High value*
	Cyan	-	
Warning indication	■ Yellow	RATIO=LL = 1	*Warning Ratio Low*
L text	**L** Black		*Text in front of the value*
L value		RATIOL	*Ratio Low value*
	Cyan	-	
OUT text	**OUT** Black		*Text in front of the line*
Warning indication	■ Yellow	OUT=HL = 1	*Warning Output High*
H text	**H** Black		*Text in front of the value*
H value		OUTPH	*Output High value*
	Black	EOLIM = 1	
	Magenta	-	
External H value		EOHL	*External Output High value*
	Magenta	EOLIM = 1	
	Black	-	
Warning indication	■ Yellow	OUT=LL = 1	*Warning Output Low*
L text	**L** Black		*Text in front of the value*
L value		OUTPL	*Output Low value*
	Black	EOLIM = 1	
	Magenta	-	
External L value		EOLL	*External Output Low value*

Description	Presentation	Condition	Remarks
	Magenta	EOLIM = 1	
	Black	-	
Man indication	■ Green	MAN = 1	*Manual mode*
Man text	**Man** Black	MAN = 1	*Text behind the indication*
Man Out text	**OUT** Black		*Text in front of the value*
Man Out value		MANOUT	*Man Out value*
	Magenta	MAN = 1	*Manual mode*
	Black	-	
Auto indication	■ Green	AUTO = 1	*Auto mode*
Auto text	**Auto** Black	AUTO = 1	*Text behind the indication*
Auto Ratio text	**RAT** Black		*Text in front of the value*
Auto Ratio value		RATIOREF	*Auto Ratio value*
	Cyan	AUTO = 1	*Auto mode*
	Black	-	
E1 indication	■ Green	E1 = 1	*E1 mode*
E1 text	**E1** Black	E1 = 1	*Text behind the indication*
E1 name		E1NAME	*User text behind the text E1*
	Green	E1 = 1	
	Black	-	
MV text	**MV** Black		*Text in front of the value*
MV value	Red	AI_ERR = 1	*AI Error*
		MV	*Measured value*
	Green	-	
MV unit		UNIT	*Measured unit*
	% Green	-	
RAT text	**RAT** Black		*Text in front of the value*
RAT value		RATIO	*Ratio value*
	Cyan	-	
RAT unit		R_UNIT	*Ratio unit*
	Cyan	-	
OUT text	**OUT** Black		*Text in front of the value*
OUT value		POUT	*Output value*
	Magenta	-	

Description	Presentation	Condition	Remarks
OUT unit		PO_UNIT	*Output unit*
	% Magenta	-	
External OUT limitation	**X** Yellow	(OUT=HL = 1 or OUT=LL= 1) and EOLIM = 1	*OUT limited by external limitation*
BIAS text	**BIAS** Black		*Text in front of the value*
BIAS value		PARAM9	*External Bias value*
	Blue	EBIAS = 1	
		BIAS	*Bias value*
	Blue	EBIAS = 0	
BIAS unit	**Ext** Blue	EBIAS = 1	*External Bias*
		PO_UNIT	*Bias units*
	% Blue	EBIAS = 0	
MV Range Max text	**MV Range Max** Black		*Text in front of the value*
MV Range Max		MAX	*Range Max of the MV value*
	Green	-	
MV Range Min text	**MV Range Min** Black	PO_UNIT	*Text in front of the value*
MV Range Min		MIN	*Range Min of the MV value*
	Green	-	

C.9 MANSTN Object Presentation (Master)

The MANSTN Object Presentation presents information for the selected MANSTN object, see Figure 3-9. MANSTN is the object type for manual controllers in the Advant OCS with Master software. MANSTN is equivalent to a PID controller in manual mode, but it is considerably easier to configure and demands less computer power during execution. MANSTN can be used, for example, to control valves for cooling water.



MAN71_8
desc of man71_8
Section 6
Class 0

MV H2 99.00
H1 80.00
L1 3.00
L2 2.00

Unacknowledge

OUT H 98.00
L 1.00

Man OUT .00 Incr
E1 kalle Op

MV .00 M/MIN
OUT 21.00 m
ACT 31.00 %

MV Range Max 100.00
MV Range Min .00
Out Range Max 89.00
Out Range Min 6.00

Count: *1
<Insert>

Figure 3-9. MANSTN Object Presentation

The operation of this presentation is described in the following table.

Description	Presentation	Condition	Remarks
HEADER			*Header*
Name		NAME	*Object name*
	Red	DISTURB = 1	*Alarm*
	Green	-	
Description		DESCR	*Description*
	Red	DISTURB = 1	*Alarm*

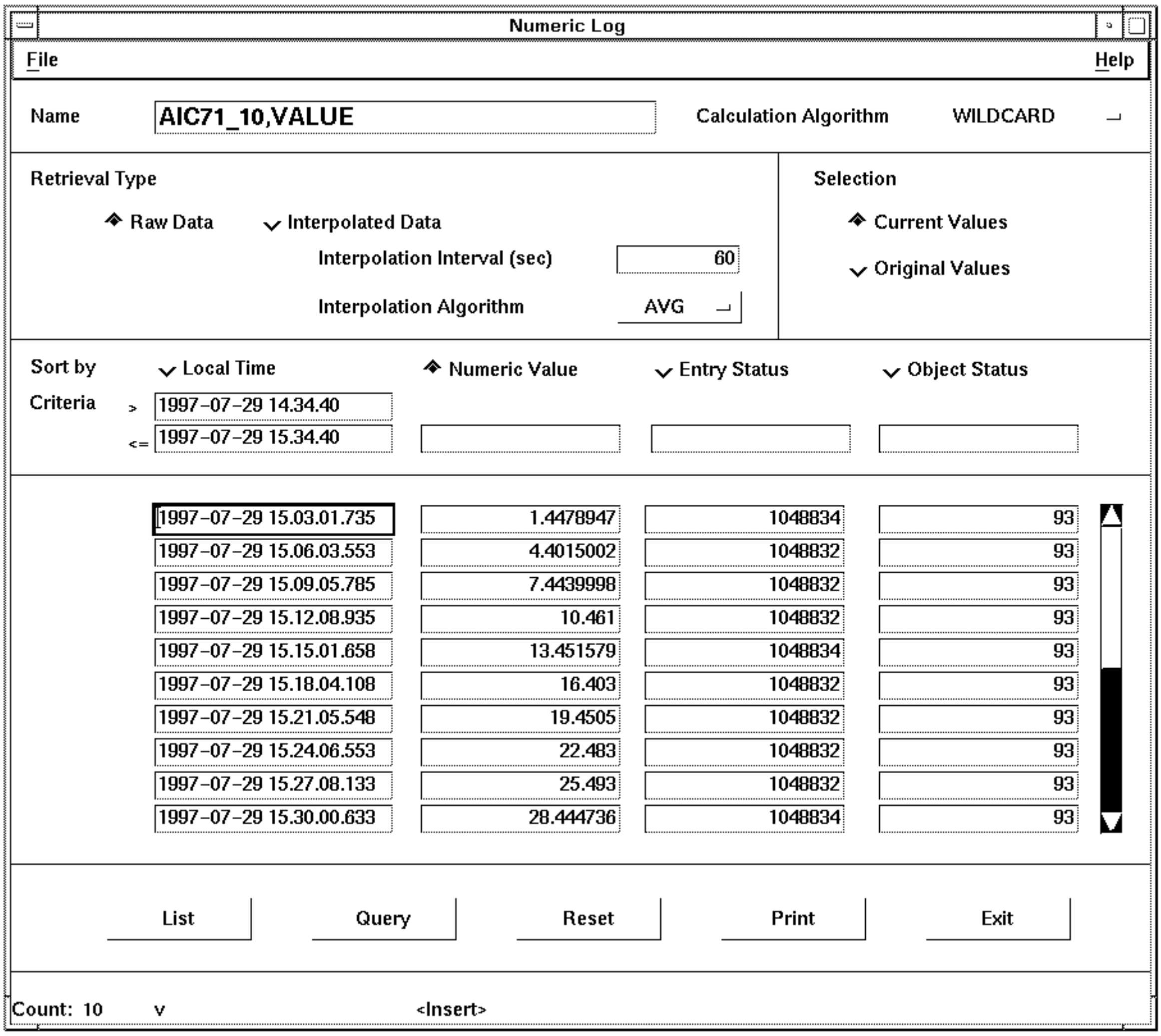
Description	Presentation	Condition	Remarks
	Green	-	
Section text	**Section** Black		*Text in front of the value*
Section value	Green	PROC_SEC	*Process section*
Class text	**Class** Black		*Text in front of the value*
Class value	Green	CLASS	*Object class*
ALARM LIMITS			*Alarm limits*
MV text	**MV** Black		*MV*
Printout blocked	**P** Yellow	PRINT_F2_BLK = 1	*Printout Blocked.*
Alarm blocked	**B** Yellow	AL_F2_BLK =1	*Alarm blocked by operator*
	Bx Yellow	AL_F2_P_BLK =1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV>H2 =1	*Alarm High limit 2*
	■ Red	MV>H2 = 1	
H2 text	**H2** Black		*Text in front of the value*
H2 value		MVH2	*High limit 2 value*
	Green	-	
Printout blocked	**P** Yellow	PRINT_F1_BLK = 1	*Printout Blocked.*
Alarm blocked	**B** Yellow	AL_F1_BLK =1	*Alarm blocked by operator*
	Bx Yellow	AL_F1_P_BLK =1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV>H1 =1	*Warning High limit 1*
	■ Red	MV>H1 = 1	
H1 text	**H1** Black		*Text in front of the value*
H1 value		MVH1	*High limit 1 value*
	Green	-	
Printout blocked	**P** Yellow	PRINT_F1_BLK = 1	*Printout Blocked.*
Alarm blocked	**B** Yellow	AL_F1_BLK =1	*Alarm blocked by operator*
	Bx Yellow	AL_F1_P_BLK =1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV<L1 =1	*Warning Low limit 1*
	■ Red	MV<L1 = 1	
L1 text	**L1** Black		*Text in front of the value*
L1 value		MVL1	*Low limit1 value*
	Green	-	
Printout blocked	**P** Yellow	PRINT_F2_BLK = 1	*Printout Blocked.*
Alarm blocked	**B** Yellow	AL_F2_BLK =1	*Alarm blocked by operator*

Description	**Presentation**	**Condition**	**Remarks**
	Bx Yellow	AL_F2_P_BLK =1	*Alarm blocked by PC-program*
Alarm indication	■ Red flashing	AU_MV<L2 =1	*Alarm Low limit 2*
	■ Red	MV<L2 = 1	
L2 text	**L2** Black		*Text in front of the value*
L2 value		MVL2	*Low limit 2 value*
	Green	-	
STATUS			*Status*
Unacknowledge indication	■ Red	AL_UNACK = !	*Unacknowledged alarm*
Unacknowledge text	**Unacknowlege** Black		*Text behind the indication*
LIMITATIONS			*Limitations*
OUT text	**OUT** Black		*Text in front of the line*
Warning indication	■ Yellow	OUT=HL = 1	*Warning Output High*
H text	**H** Black		*Text in front of the value*
H value		OUTPH	*Output High value*
	Magenta	-	
Warning indication	■ Yellow	OUT=LL = 1	*Warning Output Low*
L text	**L** Black		*Text in front of the value*
L value		OUTPL	*Output Low value*
	Magenta	-	
Man indication	■ Green	MAN = 1	*Manual mode*
Man text	**Man**		*Text behind the indication*
	Green	MAN = 1	
	Black	-	
Man Out text	**OUT** Black		*Text in front of the value*
Man Out value		MANOUT	*Man Out value*
	Magenta	MAN = 1	*Manual mode*
	Black	-	
E1 indication	■ Green	E1 = 1	*E1 mode*
E1 text	**E1** Black		*Text behind the indication*
	Green	E1 = 1	
	Black	-	
E1 name		E1NAME	*User text behind the text E1*
	Green	E1 = 1	

Description	**Presentation**	**Condition**	**Remarks**
	Black	-	
MV text	**MV** Black		*Text in front of the value*
MV value	**? ? ?** Red	ERR = 1	*Signal error*
		MV	*Measured value*
	Green	-	
MV unit		UNIT	*Measured unit*
	Green	-	
Direction	**Incr** Green	MV_DIR = INCREASING	*Increase*
	Decr Green	MV_DIR = DECREASING	*Decrease*
OUT text	**OUT** Black		*Text in front of the value*
OUT value		POUT	*Output value*
	Magenta	-	
OUT unit		PO_UNIT	*Output unit*
	% Magenta	-	
Actuator value			*The ACT object value*
ACT	**ACT** Black	CTRL = ON-OFF	*Text in front of the value*
ACT value		ACTPOS	*Actuator value*
	Blue	CTRL = ON-OFF	
ACT unit			*Actuator unit*
	% Blue	CTRL = ON-OFF	
Open or Closed	**Op** Magenta	ACT_DIR = OPENING	*Open or close information*
	Cl Magenta	ACT_DIR = CLOSING	
Ranges			*The actuator ranges are omitted since they always range from 0-100%.*
MV Range Max text	**MV Range Max** Black		*Text in front of the value*
MV Range Max vale	Green	MAX	
MV Range Min text	**MV Range Min** Black		*Text in front of the value*
MV Range Min value	Green	MIN	
Out Range Max text	**Out Range Max** Black		*Text in front of the value*
Out Range Max value	Magenta	PO_MAX	
Out Range Min text	**Out Range Min** Black		*Text in front of the value*
Out Range Min value	Magenta	PO_MIN	

C.10 NUMERIC LOG Presentation

The Numeric Log Presentation presents information for a selected Numeric Log, Figure 3-10



Name: AIC71_10,VALUE
Calculation Algorithm: WILDCARD

Retrieval Type: ◆ Raw Data ✓ Interpolated Data
Interpolation Interval (sec): 60
Interpolation Algorithm: AVG

Selection: ◆ Current Values ✓ Original Values

Sort by: ✓ Local Time ◆ Numeric Value ✓ Entry Status ✓ Object Status

Criteria > 1997-07-29 14.34.40
<= 1997-07-29 15.34.40

1997-07-29 15.03.01.735	1.4478947	1048834	93
1997-07-29 15.06.03.553	4.4015002	1048832	93
1997-07-29 15.09.05.785	7.4439998	1048832	93
1997-07-29 15.12.08.935	10.461	1048832	93
1997-07-29 15.15.01.658	13.451579	1048834	93
1997-07-29 15.18.04.108	16.403	1048832	93
1997-07-29 15.21.05.548	19.4505	1048832	93
1997-07-29 15.24.06.553	22.483	1048832	93
1997-07-29 15.27.08.133	25.493	1048832	93
1997-07-29 15.30.00.633	28.444736	1048834	93

Count: 10 v <Insert>

Figure 3-10. Numeric Log Presentation

The criteria for this presentation is described in the following table.

Calculation Algorithm	Definition of calculation algorihm (only relevant together with seamless retrieval). Possible values are AVG, MAX, MIN, INST, SUM, SUM_OF_SQRS, STANDARD_DEV, NUM_OF_VALUES or WILDCARD.
Raw Data	Raw Data returns any data for the actual time interval.

Interpolated Data	Interpolated Data returns values calculated with the specified interpolation algorithm. • Interpolation Interval (sec) Definition of interpolation interval in seconds. Note that you can specify the interval as a product, i.e 2*60*60 will result in a time interval of two hours. • Interpolation Algorithm Definition of intrpolation algorithm. Possible values are AVG, MAX, MIN or INST.
Current Values	Current will return data exactly as currently stored.
Original Values	Original will return the original data.
Local Time	Specifies the time span that will be presented. Enter year-month-date hour.minutes.seconds (1995-05-31 21.02.47). You can use % instead of one or several characters.
Numeric Value	If you leave this field empty, all numeric values are presented. You can specify a filter for numeric values as illustrated by the examples below: >**5** presents all values larger then 5 <**5** presents all values less then 5 <=**5** presents all values less then or equal to 5 **5** list of values equal to 5
Entry Status	If you leave this field empty, all entry status values are presented. You can specify a filter for entry status values as illustrated by the examples below: >**5** presents all status larger then 5 <**5** presents all status less then 5 <=**5** presents all status less then or equal to 5 **5** list of status equal to 5
Object Status	If you leave this field empty, all object status values are presented. You can specify a filter for object status values as illustrated by the examples below: >**5** presents all status larger then 5 <**5** presents all status less then 5 <=**5** presents all status less then or equal to 5 **5** list of status equal to 5

The operation of this presentation is described in the following table.

Description	Presentation	Condition	Remarks
Log Text	**Log:** Black		*Text in front of the name field*
Log name field	Black		*Complete Log name*
Calculation Algorithm text	**Calculation Algorithm:** Black		*Text in front of the selection for seamless retrieval*
Calculation Algorithm selection	AVG Black		*Calculation Algorithm values*
	MAX Black		*Calculation Algorithm values*
	MIN Black		*Calculation Algorithm values*
	INST Black		*Calculation Algorithm values*
	SUM Black		*Calculation Algorithm values*
	SUM_OF_SQRS Black		*Calculation Algorithm values*
	STANDARD_DEV Black		*Calculation Algorithm values*
	NUM_OF_VALUES Black		*Calculation Algorithm values*
	WILDCARD Black		*Calculation Algorithm values*
Retrieval Type text	**Retrieval Type** Black		*Retrieval Type*
Raw Data selection	◆ Grey filled		*Sort by Raw Data*
Raw Data text	**Raw Data** Black		*Text behind the selection button*
Interpolated Data selection	◆ Grey filled		*Sort by Interpolated Data*
Interpolated Data text	**Interpolated Data** Black		*Text behind the selection button*
Interpolation Interval (sec) text	**Interpolation Interval (sec)** Black		*Text infront of selection field*
Interpolation Interval (sec) field	Black		*Interpolation Interval parameter*
Interpolation Algorithm text	**Interpolation Algorithm** Black		*Text infront of selection field*
Interpolation Algorithm field	AVG Black		*Interpolation Algorithm values*
	MAX Black		*Interpolation Algorithm values*
	MIN Black		*Interpolation Algorithm values*
	INST Black		*Interpolation Algorithm values*
Selection text	**Selection** Black		*Selection*
Current Values selection	◆ Grey filled		*Sort by Current Values*
Current Values text	**Current Values** Black		*Text behind the selection button*
Original Values selection	◆ Grey filled		*Sort by Original Values*

Description	Presentation	Condition	Remarks
Original Values text	**Original Values** Black		*Text behind the selection button*
Sort by	**Sort by:** Black		*Sort by*
Criteria	**Criteria** Black		*Criteria*
Local Time selection	◆ Grey filled		*Sort by Local Time*
Local Time text	**Local Time** Black		*Text behind the selection button*
> text	**>** Black		*Text in front of the field*
> input field	Black		*From time filter parameter*
< text	**<** Black		*Text in front of the field*
< input field	Black		*To time filter parameter*
Numeric Value selection	◆ Grey filled		*Sort by Numeric Value*
Numeric Value text	**Numeric Value** Black		*Text behind the selection button*
Numeric Value field	Black		*Numeric Value filter parameter*
Entry Status selection	◆ Grey filled		*Sort by Entry Status*
Entry Status text	**Entry Status** Black		*Text behind the selection button*
Entry Status field	Black		*Entry Status filter parameter*
Object Status selection	◆ Grey filled		*Sort by Object Status*
Object Status text	**Object Status** Black		*Text behind the selection button*
Object Status field	Black		*Object Status filter parameter*
Scroll bar	Black		*To scroll up and down in the data fields*
Local Time data fields	Black		*Time for the presented log data*
Numeric Value data fields	Black		*Numeric value for the presented log data*
Entry Status data fields	Black		*Entry Status for the presented log data*
Object Status data fields	Black		*Status for the presented logged object*

Refer to the *AdvaInform History User's Guide* for further information on Numeric Logs.

Object Status

Refer to the *AdvaInform Object Types Reference Manual* for information about the object status.

Entry Status

Short Description	Long Description	Status
No Data: Bit(1);	No valid data. No data was collected and stored for the entry at this time. Most likely due to communications problems between History and the data source.	1
Bad Data: Bit (2);	The value stored in this entry has bad data. Most likely due to a calculation that contains No Data entries in it.	2
Entry Archived: Bit(3);	Entry is archived.	4
Value In Alarm: Bit(4);	Value is in alarm.	8
Alarm Prioirty0: Bit(5);	Alarm of priority 0.	16
AlarmPriority1: Bit (6);	Alarm of priority 1.	32
Time Quality: Bit(7);	Uncertain time quality.	64
Time Change: Bit(8);	A time change has occurred.	128
Source Time Stamp: Bit(9);	The timestamp in the entry was obtained with the value collected. Values collected by History usually use the time stamp from the source.	256
Time Stamp Added: Bit(10);	A time stamp has been added. History generated the time stamp. This is most often done on No Data entries.	512
// reserved --	Not used.	1024
//reserved --	Not used.	2048
User Entered: Bit(13);	Data was user entered. The user entered the value using the AddNumercValueEntry operation.	4096
Modified Data: Bit(14);	Data has been modified. When requesting CURRENT values from History, this is set if the user had modified the original value. If a value has been modified, then the No Data and Bad Data bits can not be set. It is assumed the modified value is correct.	8192
Modified Available: Bit(15);	Modified data exists.When requesting ORIGINAL values from History, this is set if the user had modified the original value. This indicates that there is a modified value available.	16384
//reseved --	Not used.	32768
First Entry: Bit(17);	This is the first entry. This is set when a log is activated. it does not represent a real value (No Data will also be set). It is a place holder in the log indicating when the log was activated.	65536

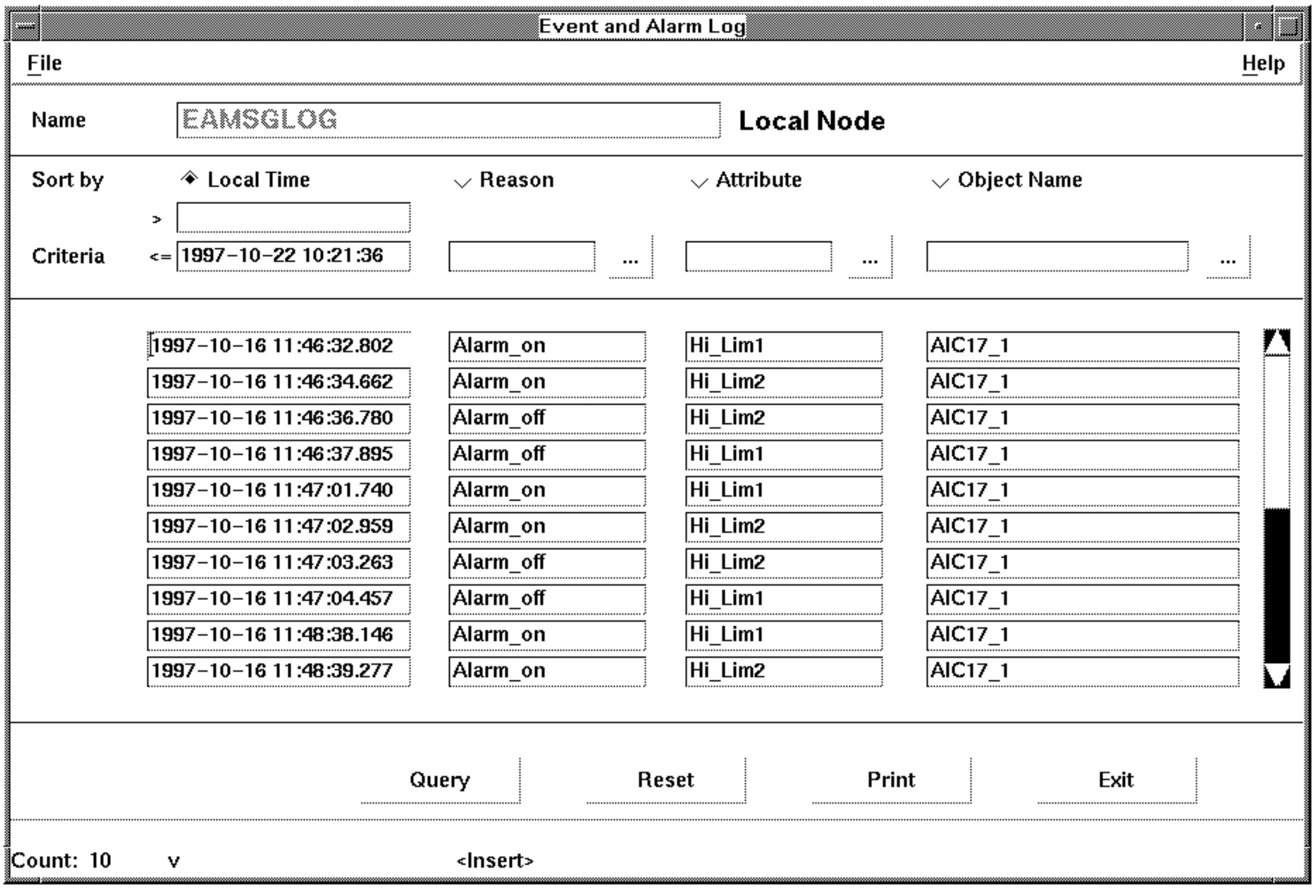
Short Description	Long Description	Status
Last Entry: Bit(18);	This is the last entry. This is set in the status of the last entry collected when a log is deactivated.	131072
Value Interpolated: Bit(19);	Data value has been interpolated. The value returned by History was interpolated during retrieval. It is not a value physically stored in the data base.	262144
//reserved--	Not used.	524288
Sequence Num0: Bit(21);	Sequence number bit 0, value 1.	1048576
Sequence Num1: Bit(22);	Sequence number bit 1, value 2.	2097152
Sequence Num2: Bit(23);	Sequence number bit 2, value 4.	4194304
Sequence Num3: Bit(24);	Sequence number bit 3, value 8.	8388608
sequence Num4: Bit(24);	Sequence number bit 4, value 16.	16777216
TTD Value: Bit(26);	Data value origins from Master TTD.	33554432
CCF Value: Bit(27);	Data value origins from MOD CCF.	67108864

C.11 Local Event Log Presentation (Master)

The Local Event Log Form is intended for presentation of local events created by local AdvaInform objects. They are stored in a history message log named EAMSGLOG. See below.

NOTE

History must be installed and you have to create the message log EAMSGLOG before any messages can be viewed.



Name EAMSGLOG **Local Node**

Sort by ◆ Local Time ◇ Reason ◇ Attribute ◇ Object Name

Criteria >

<= 1997-10-22 10:21:36

1997-10-16 11:46:32.802	Alarm_on	Hi_Lim1	AIC17_1
1997-10-16 11:46:34.662	Alarm_on	Hi_Lim2	AIC17_1
1997-10-16 11:46:36.780	Alarm_off	Hi_Lim2	AIC17_1
1997-10-16 11:46:37.895	Alarm_off	Hi_Lim1	AIC17_1
1997-10-16 11:47:01.740	Alarm_on	Hi_Lim1	AIC17_1
1997-10-16 11:47:02.959	Alarm_on	Hi_Lim2	AIC17_1
1997-10-16 11:47:03.263	Alarm_off	Hi_Lim2	AIC17_1
1997-10-16 11:47:04.457	Alarm_off	Hi_Lim1	AIC17_1
1997-10-16 11:48:38.146	Alarm_on	Hi_Lim1	AIC17_1
1997-10-16 11:48:39.277	Alarm_on	Hi_Lim2	AIC17_1

Count: 10 v

Figure C-11. Local Event Log display

The criteria for this presentation are described in the following table.

Object Name	Empty field presents all object names. Search criteria can be entered or be selected from a list of objects opened with the button beside the field. You can use % instead of one or several characters.
Reason	Empty field presents all reasons. Search criteria can be entered or be selected from a list of reasons opened with the button beside the field. You can use % instead of one or several characters.
Attribute	Empty field presents all attributes. Search criteria can be entered or be selected from a list of attributes opened with the button beside the field. You can use % instead of one or several characters.
Local Time	Specifies the time span that will be presented. Enter year-month-date hour.minutes.secunds (1995-05-31 21.02.47). You can use % instead of one or several characters.

The operation of this presentation is described in the following table.

Description	**Presentation**	**Condition**	**Remarks**
Log name field	Black		*Access name of the Message Log*
Description field			*Descriptions*
Sort by	**Sort by:** Black		*Sort by*
Criteria	**Criteria:** Black		*Criteria*
Object Name selection	◆ Grey filled		*Sort by Object Name*
Object Name text	**Object Name** Black		*Text behind the selection button*
Object Name field	Black		*Object Name filter parameter*
Object Name list	■ Grey		*List of Object Names*
Reason selection	◆ Grey filled		*Sort by Reason*
Reason text	**Reason** Black		*Text behind the selection button*
Reason field	Black		*Reason filter parameter*
Reason list	■ Grey		*List of Reasons*
Attribute selection	◆ Grey filled		*Sort by Attribute*
Attribute text	**Attribute** Black		*Text behind the selection button*
Attribute field	Black		*Attribute filter parameter*

Description	Presentation	Condition	Remarks
Attribute list	■ Grey		*List of Attributes*
Local Time selection	◆ Grey filled		*Sort by Local Time*
Local Time text	**Local Time** Black		*Text behind the selection button*
> text	**>** Black		*Text in front of the field*
> input field	Black		*From time filter parameter*
< text	**<** Black		*Text in front of the field*
< input field	Black		*To time filter parameter*
Scroll bar	Black		*To scroll up and down in the data fields*
Object Name data fields	Black		*Object name for the presented log data*
Reason data fields	Black		*Reason for the presented log data*
Property data fields	Black		*Property for the presented log data*
Local Time data fields	Black		*Local Time for the presented log data*

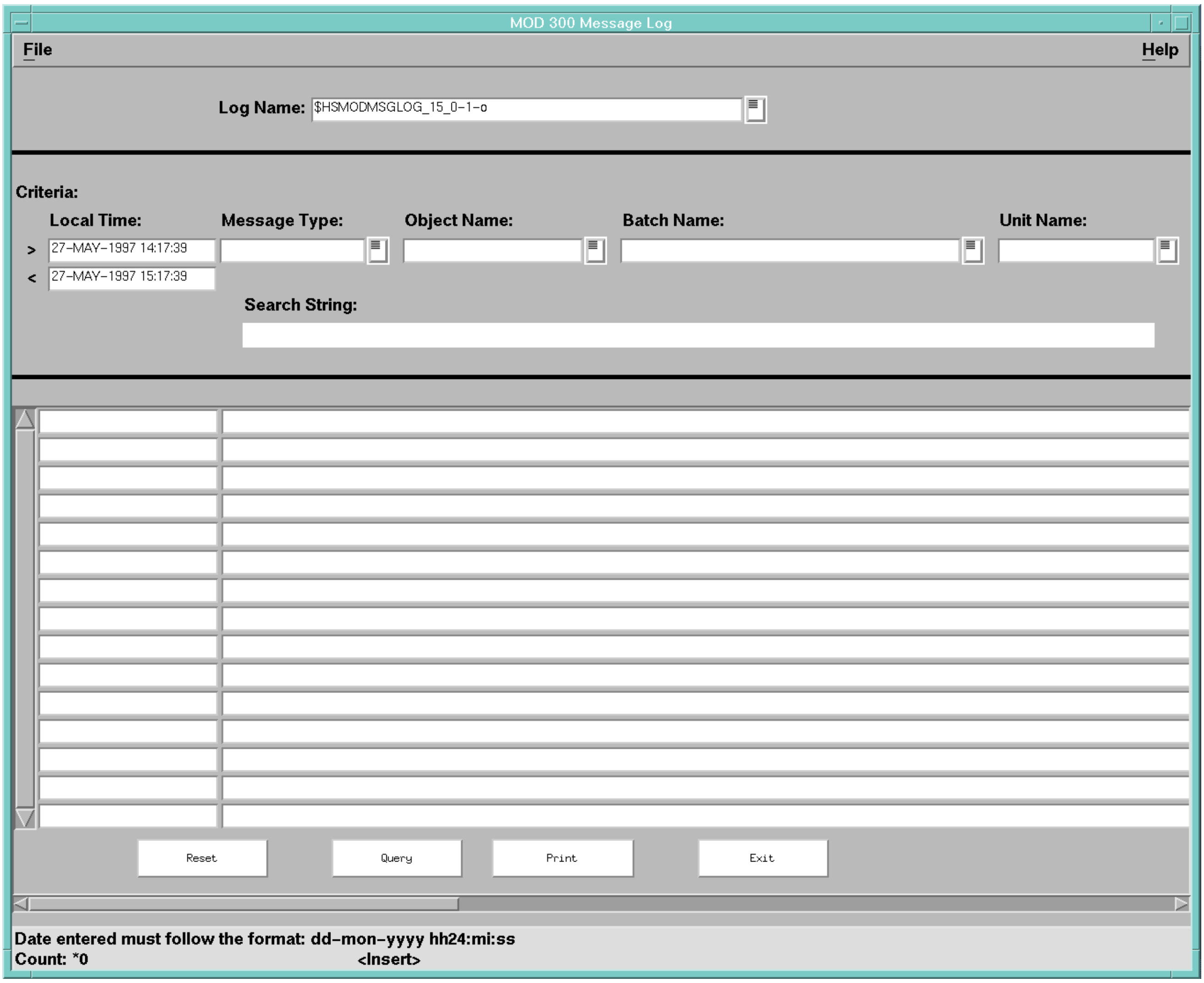
C.12 MESSAGE LOG Presentation (most applicable to MOD 300)

The Message Log object form displays information for the selected Message Log object, Figure 3-12. The Message Log object form is primarily intended for presentation of messages created by objects in controllers in the Advant OCS that generated events during a specified time period. As the History Message Logs are primarily used in MOD 300 systems most of this desciption is applicable to MOD 300 only.

To view CCF and TCL messages via the Message Log form you must create a message log with access name HCMessageCLog*Histnam* where *Histman* is the name of the History Manager for the node where the log resides. The messages are automatically stored as soon as the log is created and is activated. You are not required to add this log to the location table. Refer to the *AdvaInform History User's Guide* for additional information on MOD 300 message logs.

To use this form:

1. Choose O**bject Presentation - MOD 300 Message Log** from the **Station** menu. This displays the MOD 300 Message Log Form, Figure 3-12.



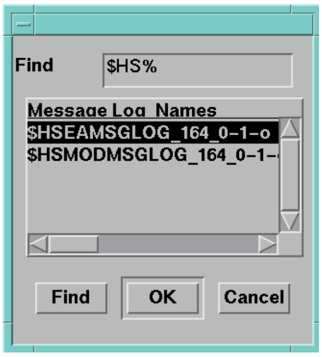
Log Name: $HSMODMSGLOG_15_0-1-o

Local Time:
> 27-MAY-1997 14:17:39
< 27-MAY-1997 15:17:39

Date entered must follow the format: dd-mon-yyyy hh24:mi:ss
Count: *0 <Insert>

Figure 3-12. MOD 300 Message Log Form

2. Select a log name. Click on the corresponding button to display a list of log names, Figure 3-13.



Find $HS%

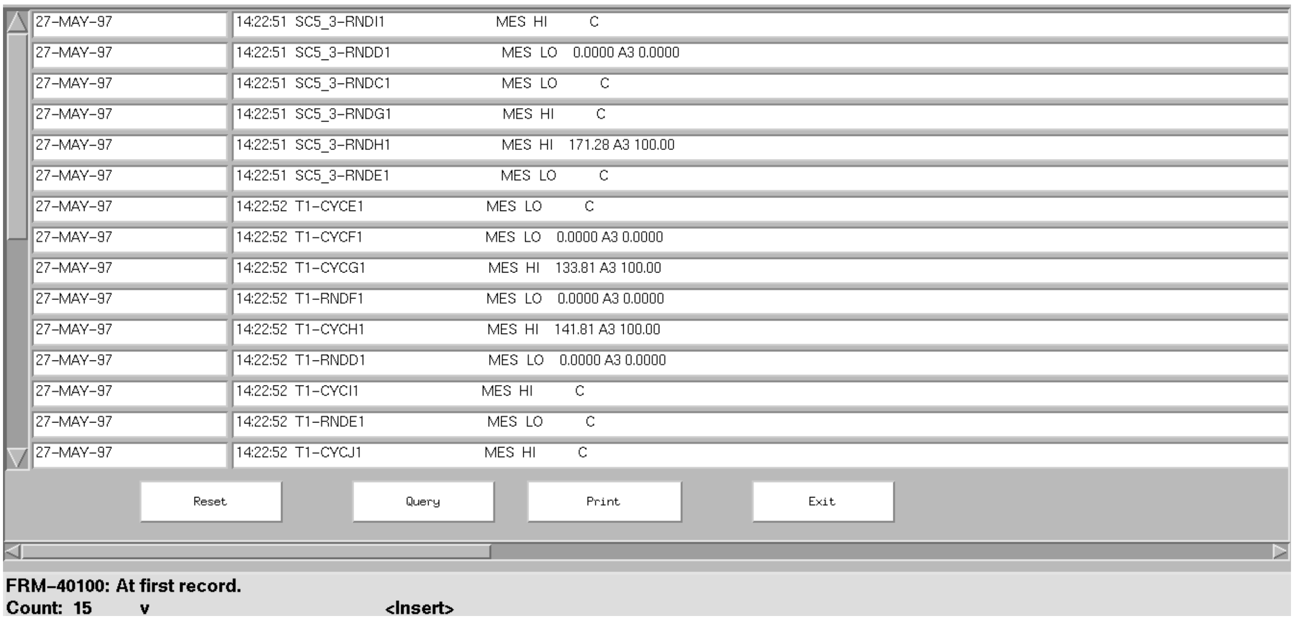
Message Log Names
$HSEAMSGLOG_164_0-1-o
$HSMODMSGLOG_164_0-1-

Figure 3-13. Log Name Selection Dialog

3. Enter the selection criteria. You can specify a date and time range, message type, tag name, batch name, and unit name. The default criteria is all messages. Guidelines for specifying one or more of these selection criteria are provided below:

Local Time	The default is all messages up to the current time. To enter a specific range, enter a starting date and time in the top (>) field, and an ending date and time in the bottom (<) field. The format is: dd-mon-yyyy hh24:mi:ss.
Message Type	The default is all message types. To enter a specific message type, you can either make an entry directly in the Message Type field, or click on the corresponding button to display a list of possible message types. The choices are: ALARM, BILLBOARD, PARAMETER_CHANGE, LADDER_LOGIC, SPC, ALARM_ACKNOWLEDGE.
Object Name	The default is all objects. To enter a specific object, you can either enter the name directly in the Object Name field, or click on the corresponding button to display a list of possible object names.
Batch Name	The default is all batches. To enter a specific batch, you can either enter the batch name directly in the Batch Name field, or click on the corresponding button to display a list of possible batch names.
Unit Name	The default is all units. To enter a specific unit, you can either enter the unit name directly in the Unit Name field, or click on the corresponding button to display a list of possible unit names.
Search String	You can use this field to enter a search string to find a specific object in the list resulting from the query.

4. When you are finished with the selection criterial, click the **QUERY** button to display the messages,Figure 3-14 .



```
27-MAY-97    14:22:51  SC5_3-RNDI1        MES  HI        C
27-MAY-97    14:22:51  SC5_3-RNDD1        MES  LO    0.0000 A3 0.0000
27-MAY-97    14:22:51  SC5_3-RNDC1        MES  LO        C
27-MAY-97    14:22:51  SC5_3-RNDG1        MES  HI        C
27-MAY-97    14:22:51  SC5_3-RNDH1        MES  HI    171.28 A3 100.00
27-MAY-97    14:22:51  SC5_3-RNDE1        MES  LO        C
27-MAY-97    14:22:52  T1-CYCE1           MES  LO        C
27-MAY-97    14:22:52  T1-CYCF1           MES  LO    0.0000 A3 0.0000
27-MAY-97    14:22:52  T1-CYCG1           MES  HI    133.81 A3 100.00
27-MAY-97    14:22:52  T1-RNDF1           MES  LO    0.0000 A3 0.0000
27-MAY-97    14:22:52  T1-CYCH1           MES  HI    141.81 A3 100.00
27-MAY-97    14:22:52  T1-RNDD1           MES  LO    0.0000 A3 0.0000
27-MAY-97    14:22:52  T1-CYCI1           MES  HI        C
27-MAY-97    14:22:52  T1-RNDE1           MES  LO        C
27-MAY-97    14:22:52  T1-CYCJ1           MES  HI        C
```

```
FRM-40100: At first record.
Count:  15      v                 <Insert>
```

Figure 3-14. Example, Query Result

5. You can use the **RESET** button to clear the form, or the **PRINT** button to print the contents of the form.

6. To exit, either click on the **EXIT** button, or choose **Exit** from the **File** menu.

The operation of this presentation is described in the following table.

Description	**Presentation**	**Condition**	**Remarks**
Log name field	Black		*Complete Log name*
Sort by	**Sort by:** Black		*Sort by*
Criteria	**Criteria:** Black		*Criteria*
Local Time selection	◆ Grey filled		*Sort by Local Time*
Local Time text	**Local Time** Black		*Text behind the selection button*
> text	**>** Black		*Text in front of the field*
> input field	Black		*From time filter parameter*
< text	**<** Black		*Text in front of the field*
< input field	Black		*To time filter parameter*
Message Type selection	◆ Grey filled		*Sort by Message Type*

Description	Presentation	Condition	Remarks
Message Type text	**Message Type** Black		*Text behind the selection button*
Message Type field	Black		*Message Type filter parameter*
Message Type list	■ Grey		*List of Message Types*
Message selection	◆ Grey filled		*Sort by Message*
Message text	**Message** Black		*Text behind the selection button*
Message field	Black		*Message filter parameter*
Scroll bar	Black		*To scroll up and down in the data fields*
Local Time data fields	Black		*Time for the presented log data*
Message Type data fields	Black		*Message Type for the presented log data*
Message data fields	Black		*Message for the presented log data*

Index

Numerics
3-GL (third generation language) 1-10, 1-31

A
access name 4-6
AdvaBuild Central Backup 4-51
AdvaBuild Object Type Builder 1-29
AdvaBuild Options 4-51
AdvaInform Basic Functions 1-1, 3-1
AdvaInform Basic Objects 1-11
AdvaInform Basic Software 1-11
AdvaInform Calculation 4-53
AdvaInform History 4-53
AdvaInform relational database 1-15
    log into 3-1
AdvaInform Reports 1-21
AdvaInform software
    shutdown 3-17
    start-up 3-17
    supervision 3-17
AdvaInform SQL*Connect 4-52
AdvaInform SQL*Connect Server 1-21
AdvaInform User API 1-10, 1-15, 1-29, 4-50
Advant Station 500 OS 3-12
Advant Station 500 Series 1-1 to 1-2
AI object presentation C-1
alarm acknowledge 1-28
Alarm and Event Distributor - Process bceadistr
    error messages 5-3
    information messages 5-6
    warning messages 5-5
Alarm and Event Handling 1-28
alarm messages 5-7
AO object presentation C-5
API 1-10
application 3-17
Attribute 1-10

B
backup 1-31
Basic Object 1-19, 1-29, C-40

C
C 1-31
C and C++ developers licenses 1-32
change 4-55
    language 4-46
    password 4-54 to 4-55
Command 1-10
communication 1-19, 3-16
Communication Set-up 3-16
Composite Attributes 1-10
Configuration of SQL*Net C-1
control network 1-10, 1-19
conventions 1-4
core system 1-10, 1-14, 1-19
CREATE B-3
create
    database link B-14
create view
    SQL lanugage reference B-3
current user 4-54

D
DAT object presentation C-13
Database Access 3-8
    window 4-5
database links 1-20
database login over SQL*NET 3-8
database theory
    hierarchical database A-1
    network database A-1
    relational database A-1
date and time setting 3-17
DCS 1-10
default language 4-46
define system variables 3-17
delete SQL language reference B-26
DESCRIBE 4-6
describe SQL language reference B-6
DI object presentation C-8
dimension error log 3-17
DO object presentation C-11
dynamic function keys 3-12

E

EAMSGLOG 4-10, C-40
editor 1-28, 3-17
embedded SQL 1-22
environment 3-4
error message 5-3
ES 1-10
event
    message 1-29
event and alarm distributor object 1-29
external node 1-11, 1-19 to 1-20

F

fault finding and repair 5-9

H

hard disk
    drives 3-17
hardware
    setup 2-1
Help menu 4-55
History 4-53
History logs 1-12
home directory 3-4
hosts file 3-12
HP CDE 1-23
HP Common Desktop Environment (HP CDE)
    exit 3-7
    front panel 3-6
    login screen 3-5
HP-elm 4-26
HP-UX
    log into 3-5
    users in Advant Station IMS 3-4

I

IMS 1-1, 1-11, 3-2
IMS menu 1-11, 1-23
    AdvaBuild
        Central Backup 4-51
        Object Type Builder 4-51

I (continued)

    AdvaInform
        Calculations 4-53
        Event Log 4-53
        History 4-53
        Object Handling 4-52
        PDL 4-54
        Reports 4-53
        SPC 4-54
        SQL*Connect 4-52
        User API 4-53
    AdvaTalk
        Link to MATLAB 4-54
    File
        Exit 4-47
    Help 4-55
        On Copyrights 4-56
        On Help 4-56
        On Index 4-56
        On Keys 4-56
        On Options 4-56
        On Version 4-56
        On Window 4-56
    icon 3-6
    items 3-2
    log into 3-1, 3-5
    log out 3-7
    open 3-6
    privileges 3-1
    Session
        Language 4-55
        Password 4-55
        Show User 4-55
    Station
        Database Access 4-49
        Editor 4-50
        Mail 4-50
        Object Presentation 4-49
        Object Reference Maintenance 4-50
        System Message 4-50
        System Status 4-48
IMS_TABLES 4-6
IMSmenu command 3-6 to 3-7, 5-3
    start-up 3-5
IMSstart command 5-3
information
    messages 5-6, 5-8

K
ksh 4-49

L
LANG 4-46
Language
    support 4-55
Local Event Log 4-10
log in 3-1, 3-5
    file 3-5
log name 4-6
    syntax 4-7
log out 3-7

M
mail 1-28, 3-17
maintenance 3-17, 5-1
MANSTN object presentation C-30
MasterAid 3-17
MasterView 1-28
menu items 1-24
message box 5-2
Message Log 1-29
    object presentation 1-29, C-40
multiple views
    database theory A-2
multi-user 1-23

N
native language support 4-46
network
    parameters 3-16
NLS_LANG 4-46
Number of concurrent database users 1-16
Numeric Log object 4-6
    presentation C-34

O
Object Presentation 1-27, 4-10
Object Type Builder 1-29, 4-51
object types
    AdvaInform Basic Objects 1-29
    AdvaInform User Objects 1-29
Objects 1-12
ocsmgr 3-16
ocsmgr HP-UX users 3-4
oerr on-line help 5-2
Off-Line Configuration 1-12

O (continued)
On-Line Configuration 1-12
operating system
    user 3-16
Operator station 1-28
ORA number 5-2
Oracle 1-32
    error message 5-2
    forms 1-32
    HP-UX users 3-4
    RDBMS 1-15
    Run-Time licence 4-25
OS 1-12

P
password 3-1
    change 4-54
peripheral devices 2-1
PIDCON object presentation C-18
ping tool 3-12
PL/SQL 4-49
    database theory A-3
Plant Network 1-1, 1-13, 1-16, 1-19 to 1-20
pre-defined
    user 3-16
printer 3-17
privilege 3-1
Process Display 3-12
Process Startup 3-17
Programmer 2000 - Pro*C 4-51

Q
Query
    message log 4-8

R
RATIOSTN object presentation C-25
RDBMS 1-13
RDBMS Kernel
    database theory A-2
real time 1-19
Real-Time Accelerator Board 1-19
Relational Database 1-15, A-1
remote
    host 3-12
Remote Oracle 1-13
rhosts 3-12
Rollback segments B-22
root 3-17

R (continued)
root/superuser
  HP-UX users 3-4
row
  database theory A-2
RTA 1-13, 1-19
  Board Configuration 3-17
  settings 3-17

S
SAM 3-17
SELECT 4-6, 4-25
  database theory A-3
  SQL language reference B-7, B-11
select statement clauses
  SQL language reference B-13
Session menu 4-54
Set Language 4-46
Settings menu 3-15
severity code 5-1
Show User 4-45
shutdown 3-17
SQL 1-13, 1-15, 4-49, A-2
  communications 1-20
  database theory A-3
SQL language reference B-1
  CREATE VIEW B-3
  DELETE B-26
  DESCRIBE B-6
  INSERT B-24
  SELECT B-7, B-11
  select statement clauses B-13
  UPDATE B-21
SQL query
  execution of 4-5
SQL*Connect 4-52
SQL*Connect Server 1-21
SQL*Net 1-14, 1-16, 1-20
SQL*Plus 4-49
start-up 3-17
Station Backup 1-31, 3-16
Station menu 4-48
Subscription 1-13
superuser 3-17
system administrator 3-17
System Message 3-17, 5-1
  display 5-1
  Examine Window 4-28
System Status 1-3, 1-26

S (continued)
System Status display
  alarm messages 5-7
  information messages 5-8
  warning messages 5-7
System Variables 3-17
  edit 3-17
  settings 3-17

T
table
  database theory A-2
TCP/IP 1-20, 1-23, 5-3
terminal window 1-28, 4-49
terminology 1-10
text editor 1-28
TEXT object presentation C-15
time synchronization 1-30, 3-10
Time Tagged Data 3-13
TO_DATE function 4-6
trace files 5-2
TTD Configuration 3-13, 4-53
  tool 3-13

U
update
  SQL language reference B-21
user 1-23
  authority 3-16
  categories 3-1
  interface 1-23
  name 3-1
  privileges 3-2
User Object Type 1-14
User Objects 1-19, 1-29

V
view
  database theory A-2

W
warning messages 5-5, 5-7

X
X-
  client 1-23
  host 3-12
  terminal 1-23

3BSE 005 992R0301

October 1997